



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 2000082003 A

(43) Date of publication of application: 21 . 03 . 00.

(51) Int. Cl. G06F 12/00
G06F 13/00

(21) Application number: 11257413

(22) Date of filing: 05 . 07 . 95

(62) Division of application: 07170019

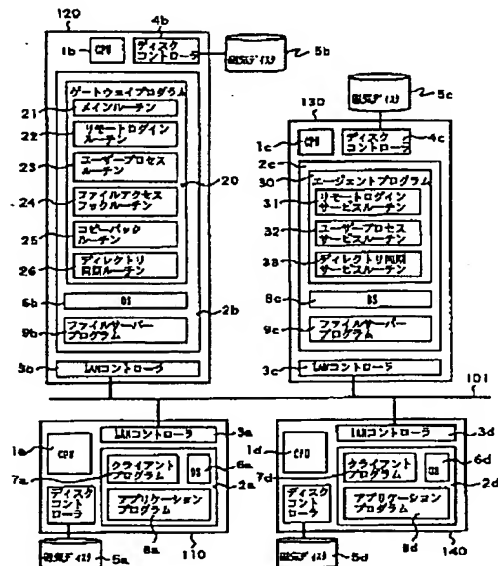
(71) Applicant: HITACHI LTD HITACHI
SOFTWARE ENG CO LTD(72) Inventor: ITO HIROMICHI
ARAI MASATO
NAKADA YUKIO
ITO HISAYA
MORI MITSURU(54) INFORMATION PROCESSING SYSTEM
ENABLING ACCESS TO DIFFERENT KIND OF
FILE AND CONTROL METHOD THEREOF

(57) Abstract:

PROBLEM TO BE SOLVED: To provide an information processing system which makes it possible to efficiently access files on plural file servers having different file managing methods by a single client program and to provide the control method thereof.

SOLUTION: A copy of the directory structure of a file on a 2nd server information processor is generated on a 1st server information processor 120 by providing a gateway program 20 on the 1st server information processor 120 and an agent program 30 to be operated corresponding to the gateway program 20 on the 2nd server information processor 130 and when the copy is generated, the data in the file are not copied, but only when the file is opened from an application program 8a of a client information processor 110, the data are copied.

COPYRIGHT: (C)2000,JPO



(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号
特開2000-82003
(P2000-82003A)

(43) 公開日 平成12年3月21日 (2000.3.21)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 0 6 F 12/00	5 2 0	G 0 6 F 12/00	5 2 0 J
	5 3 3		5 3 3 J
13/00	3 5 1	13/00	3 5 1 E

審査請求 未請求 請求項の数28 O L (全 29 頁)

(21) 出願番号 特願平11-257413
(62) 分割の表示 特願平7-170019の分割
(22) 出願日 平成7年7月5日 (1995.7.5)

(71) 出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地
(71) 出願人 000233055
日立ソフトウェアエンジニアリング株式会
社
神奈川県横浜市中区尾上町6丁目81番地
(72) 発明者 伊藤 浩道
神奈川県川崎市麻生区王禅寺1099番地 株
式会社日立製作所システム開発研究所内
(74) 代理人 100087170
弁理士 富田 和子

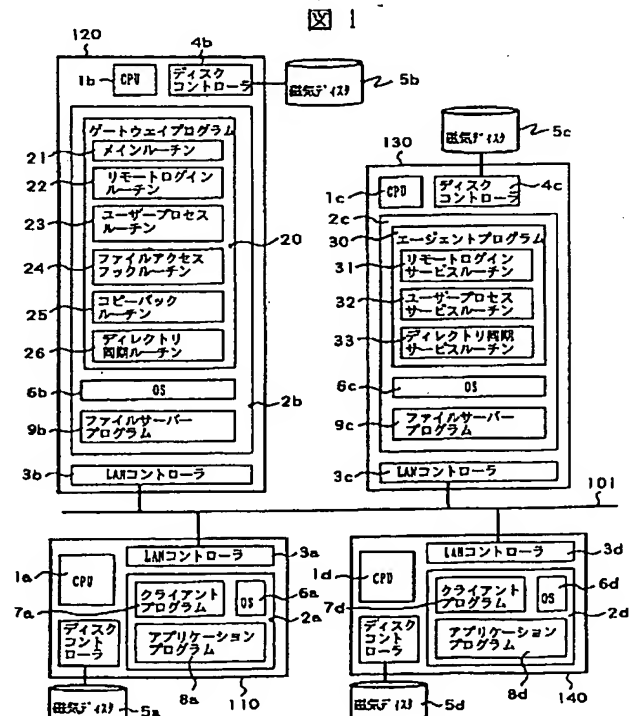
最終頁に続く

(54) 【発明の名称】 異種ファイルへのアクセスを可能とする情報処理システム及びその制御方法

(57) 【要約】

【課題】 単一のクライアントプログラムにより、ファイル管理方法が異なる複数のファイルサーバ上のファイルを効率よくアクセスすることが可能な情報処理システム及びその制御方法を提供する。

【解決手段】 第一のサーバ情報処理装置120上にゲートウェイプログラム20を設け、該ゲートウェイプログラム20に対応して動作するエージェントプログラム30を第二のサーバ情報処理装置130上に設けることにより、第二のサーバ情報処理装置上のファイルのディレクトリ構造のコピーを第一のサーバ情報処理装置120上に作成すると共に、当該コピーを作成する時点ではファイルの中のデータはコピーせず、クライアント情報処理装置110のアプリケーションプログラム8aから該ファイルがオープンされたときに限りデータのコピーを行う。



【特許請求の範囲】

【請求項 1】ファイルの共用を可能とする第一のファイルサーバプログラムを具備する第一のサーバ情報処理装置と、前記第一のファイルサーバプログラムとは、共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置と、前記第一及び第二のサーバ情報処理装置を接続するネットワークとを備える情報処理システムであって、前記第一のサーバ情報処理装置は、前記第二のサーバ情報処理装置が管理するファイル（以下、異種ファイルと呼ぶ）へのアクセス処理を実行するアクセス手段を備え、前記第二のサーバ情報処理装置は、前記アクセス手段から送られてくる処理要求を受け入れて、該処理要求に対応する処理を実行するサービス手段を備え、前記アクセス手段は、前記サービス手段を介して前記異種ファイルのディレクトリ構造の少なくとも一部の構造に関する情報を検出して、該検出した情報に応じて、前記第一のファイルサーバプログラムが管理することができる形態を備えた、前記異種ファイルのディレクトリ構造のコピーを作成し、該作成した異種ファイルのディレクトリ構造のコピーを、前記第一のファイルサーバプログラムが管理するファイルのディレクトリ構造中に追加すると共に、前記情報の検出を所定のタイミングで実行し、その度に異種ファイルのディレクトリ構造のコピーを更新するディレクトリ同期手段を備えることを特徴とする情報処理システム。

【請求項 2】請求項 1 記載の情報処理システムにおいて、前記ディレクトリ同期手段は、前回の同期時刻を記憶する記憶手段と、前記コピーの対象となっている、前記ディレクトリ構造の少なくとも一部に含まれる異種ファイルのうち、前記記憶されている前回の同期時刻以降に変更のあったファイルを前記サービス手段を介して検索する検索手段と、前記検索されたファイルについて、前記異種ファイルのディレクトリ構造のコピーを更新する更新手段とを備えることを特徴とする情報処理システム。

【請求項 3】請求項 1 記載の情報処理システムにおいて、前記ディレクトリ同期手段は、前記異種ファイルのディレクトリ構造のコピー作成に際して、コピーされる異種ファイルがディレクトリファイル以外の通常ファイルの場合、当該異種ファイルに格納されているデータをコピーしないことを特徴とする情報処理システム。

【請求項 4】請求項 1 記載の情報処理システムにおいて、前記ディレクトリ同期手段は、前記異種ファイルのディレクトリ構造のコピー作成に際して、コピーする異種ファイルがディレクトリファイル以外の通常ファイルの場合、作成されるファイルは、当該異種ファイルの最後のデータ位置と同じ位置にだけダミーデータを書き込んだスパースファイルであることを特徴とする情報処理システム。

【請求項 5】請求項 1 記載の情報処理システムにおいて、前記ディレクトリ同期手段は、作成した前記異種ファイルのディレクトリ構造のコピーが占有する記憶領域の容量が、コピー元のディレクトリ構造が占有している記憶領域の容量よりも少くなるように、前記異種ファイルのディレクトリ構造のコピーを作成することを特徴とする情報処理システム。

【請求項 6】請求項 1 記載の情報処理システムにおいて、前記ディレクトリ同期手段は、前記異種ファイルのディレクトリ構造のコピー作成に際して、作成した前記異種ファイルのディレクトリ構造のコピーに含まれるファイルの日付及びサイズを、コピー元の異種ファイルと同じとすることを特徴とする情報処理システム。

【請求項 7】ファイルの共用を可能とする第一のファイルサーバプログラムを具備する第一のサーバ情報処理装置と、前記第一のファイルサーバプログラムとは、共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置と、前記第一のファイルサーバプログラムを介して前記第一のサーバ情報処理装置上の共用ファイルへアクセスするクライアントプログラムを具備する少なくとも 1 台のクライアント情報処理装置と、前記第一のサーバ情報処理装置及び前記少なくとも 1 台のクライアント情報処理装置を接続する第一のネットワーク、及び、前記第一及び第二のサーバ情報処理装置を接続する第二のネットワークとを備える情報処理システムであって、前記第一のサーバ情報処理装置は、前記第二のサーバ情報処理装置が管理する異種ファイルへのアクセスを行うアクセス手段を備え、前記第二のサーバ情報処理装置は、前記アクセス手段と対応して動作するサービス手段を備え、前記アクセス手段は、前記各クライアントプログラムから送られてくるファイルへの処理要求を受け入れ、該処理要求が前記第二のファイルサーバプログラムが管理するファイル（以下、異種ファイルと呼ぶ）への処理要求を含むかどうかを判断し、それを含むと判断された場合には、前記処理要求に含まれている異種ファイルへの処理要求を、前記第二のファイルサーバプログラムで利用できる第二の処理要求に変換し、該変換した第二の処理要求を前記第二のサーバ情報処理装置へ送るものであり、前記サービス手段は、前記第一のサーバ情報処理装置から送られてくる、前記第二の処理要求を受け入れ、それに対応して予め定められている処理を実行することを特徴とする情報処理システム。

【請求項 8】請求項 7 記載の情報処理システムにおいて、前記アクセス手段は、前記各クライアントプログラムから送られてくる、前記異種ファイルへのアクセスをフックするフック手段と、前記アクセスが前記異種ファイルへのオープン要求であった場合に、該当するファイルのデータを、前記サービス手段を介して前記第一のファイルサーバプログラムが管理するディレクトリ構造内

へコピーするファイルコピー手段と、前記ファイルコピー手段でコピーしたファイルを、前記サービス手段を介して前記第二のサーバ情報処理装置へ書き戻すコピーバック手段と、前記アクセスが前記異種ファイルへのクローズ要求であった場合に、前記コピーバック手段を別プロセスとして起動するコピーバック起動手段とを備えることを特徴とする情報処理システム。

【請求項 9】請求項 8 記載の情報処理システムにおいて、前記アクセス手段は、前記クライアントプログラムからの前記異種ファイルへのクローズ要求を、前記コピーバック手段によるコピーバック処理の完了を待つことなく完了させ、前記クライアントプログラムからの前記クローズ要求に続く処理を続けて受け入れることを特徴とする情報処理システム。

【請求項 10】請求項 8 記載の情報処理システムにおいて、前記ファイルコピー手段は、前記ファイルのデータのコピーが既に存在するかどうかを調べ、存在する場合には、前記コピー処理を行わないことを特徴とする情報処理システム。

【請求項 11】請求項 8 記載の情報処理システムにおいて、前記コピーバック手段は、コピーバック処理終了後、前記ファイルコピー手段でコピーしたファイルのデータを消去することを特徴とする情報処理システム。

【請求項 12】請求項 11 記載の情報処理システムにおいて、前記コピーバック手段は、当該ファイルに対するクローズ要求に引き続き、当該ファイルに対するオープン要求が発行されているときは、コピーバック処理終了後の当該ファイルのデータの消去を行わないことを特徴とする情報処理システム。

【請求項 13】請求項 8 記載の情報処理システムにおいて、前記コピーバック手段は、コピーバック処理の開始を予め定めた時間だけ遅延し、この遅延時間内に、当該ファイルに対するクローズ要求に引き続き、当該ファイルに対する削除要求が発行された場合は、コピーバック処理を実行しないことを特徴とする情報処理システム。

【請求項 14】請求項 13 記載の情報処理システムにおいて、前記コピーバック手段が、前記遅延時間内に、前記クローズ要求に引き続き、当該ファイルへのオープン要求が発行された場合は、当該遅延を中止し、直ちにコピーバック処理を実行することを特徴とする情報処理システム。

【請求項 15】請求項 8 記載の情報処理システムにおいて、前記コピーバック手段は、当該ファイルへのライトモードでのオープン要求が他に行われていない場合にだけコピーバック処理を行うことを特徴とする情報処理システム。

【請求項 16】請求項 11 記載の情報処理システムにおいて、前記コピーバック手段は、当該クローズ要求に対応するオープン要求以外に、当該ファイルに対するオープン要求が発行されているときは、コピーバック処理終

了後のデータ消去を行わないことを特徴とする情報処理システム。

【請求項 17】請求項 11 記載の情報処理システムにおいて、前記データの消去は、当該ファイルを、コピー元のファイルの最後のデータ位置と同じ位置にだけダミーデータを書き込んだスパースファイルとすることによって行うことを特徴とする情報処理システム。

【請求項 18】請求項 8 記載の情報処理システムにおいて、前記コピーバック手段は、前記コピーしたファイルの属性のうち、当該ファイルに格納されているデータに対しての修正処理の有無を示す属性を検査し、当該属性が前記データの修正を示していた場合にだけコピーバック処理を行うことを特徴とする情報処理システム。

【請求項 19】請求項 18 記載の情報処理システムにおいて、前記属性が、ファイルのバックアップの必要性を示すアーカイブビットであることを特徴とする情報処理システム。

【請求項 20】請求項 8 記載の情報処理システムにおいて、前記ファイルコピー手段は、ファイルコピー処理に先だって前記サービス手段を介して前記異種ファイルへのオープン処理を行うものであり、当該オープン処理が、前記第二のサーバプログラムでの同時オープン可能なファイル数の制限を越えたためにエラーとなった場合には、当該オープン処理を成功するまでリトライすることを特徴とする情報処理システム。

【請求項 21】ファイルの共用を可能とする第一のファイルサーバプログラムを具備する第一のサーバ情報処理装置と、前記第一のファイルサーバプログラムとは、共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置と、前記第一のファイルサーバプログラムを介して前記第一のサーバ情報処理装置上の共用ファイルへアクセスするクライアントプログラムを具備する少なくとも 1 台のクライアント情報処理装置と、前記第一のサーバ情報処理装置及び前記少なくとも 1 台のクライアント情報処理装置を接続する第一のネットワーク、及び、前記第一及び第二のサーバ情報処理装置を接続する第二のネットワークとを備える情報処理システムであって、前記第一のサーバ情報処理装置は、前記第二のサーバ情報処理装置が管理するファイル（以下、異種ファイルと呼ぶ）へのアクセスを行うアクセス手段を備え、前記第二のサーバ情報処理装置は、前記アクセス手段と対応して動作するサービス手段を備え、前記アクセス手段は、前記各クライアントプログラムから送られてくる、前記異種ファイルへのアクセスをフックするフック手段と、前記アクセスが前記異種ファイルへのオープン要求であった場合に、該当するファイルのデータを、前記サービス手段を介して前記第一のファイルサーバプログラムが管理するディレクトリ構造内へコピーするファイルコピー手段と、前記ファイルコピー手段でコピーしたファイ

ルを、前記サービス手段を介して、前記コピー元へ書き戻すコピーバック手段と、前記アクセスが前記異種ファイルへのクローズ要求であった場合に、前記コピーバック手段を別プロセスとして起動するコピーバック起動手段と、前記サービス手段を介して前記異種ファイルのディレクトリ構造の少なくとも一部に関する情報を検出して、該検出した情報に応じて、前記第一のファイルサーバプログラムが管理することができる形態を備えた、前記異種ファイルのディレクトリ構造のコピーを作成し、該作成した異種ファイルのディレクトリ構造のコピーを、前記第一のファイルサーバプログラムが管理するファイルのディレクトリ構造中に追加すると共に、前記情報の検出を周期的に行い、その度に前記異種ファイルのディレクトリ構造のコピーを更新するディレクトリ同期手段とを備えることを特徴とする情報処理システム。

【請求項22】ファイルの共用を可能とする第一のファイルサーバプログラムを具備する第一のサーバ情報処理装置と、前記第一のファイルサーバプログラムとは、共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置と、前記第一のファイルサーバプログラムを介して前記第一のサーバ情報処理装置上の共用ファイルへアクセスするクライアントプログラムを具備する少なくとも1台のクライアント情報処理装置と、前記第一のサーバ情報処理装置及び前記少なくとも1台のクライアント情報処理装置を接続する第一のネットワーク、及び、前記第一及び第二のサーバ情報処理装置を接続する第二のネットワークとを備える情報処理システムの、異種ファイルアクセスを可能とする制御方法であって、前記異種ファイルのディレクトリ構造の少なくとも一部の構造に関する情報を検出して、該検出した情報に応じて、前記第一のファイルサーバプログラムが管理することができる形態を備えた、前記第二のサーバ情報処理装置が管理するファイルのディレクトリ構造のコピーを作成し、該作成したディレクトリ構造のコピーを、前記第一のファイルサーバプログラムが管理するファイルのディレクトリ構造中に追加する処理と、前記情報の検出を所定のタイミングで実行し、その度に前記異種ファイルのディレクトリ構造のコピーを更新する処理とを含むことを特徴とする情報処理システムの制御方法。

【請求項23】ファイルの共用を可能とする第一のファイルサーバプログラムを具備する第一のサーバ情報処理装置と、前記第一のファイルサーバプログラムとは、共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置と、前記第一のファイルサーバプログラムを介して前記第一のサーバ情報処理装置上の共用ファイルへアクセスするクライアントプログラムを具備する少なくとも1台のクライアント情報処理装置と、前記第一のサーバ情報処理装置及び前記少なくとも1台のクライ

ト情報処理装置を接続する第一のネットワーク、及び、前記第一及び第二のサーバ情報処理装置を接続する第二のネットワークとを備える情報処理システムの異種ファイルへのアクセスを可能とする制御方法であって、前記クライアントプログラムから送られてくる、前記第二のサーバ情報処理装置が管理するファイル（以下、異種ファイルと呼ぶ）へのアクセスをフックする処理と、前記アクセスが前記異種ファイルへのオープン要求であった場合に、該当するファイルのデータを、前記第一のファイルサーバプログラムが管理するディレクトリ構造内へコピーする処理と、前記ファイルコピー手段でコピーしたファイルを、前記サービス手段を介して、前記第二のサーバ情報処理装置に書き戻す処理と、前記アクセスが前記異種ファイルへのクローズ要求であった場合に、前記コピーバック手段を別プロセスとして起動する処理とを含むことを特徴とする情報処理システムの制御方法。

【請求項24】ファイルの共用を可能とする第一のファイルサーバプログラムを具備し、前記第一のファイルサーバプログラムとは共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置とネットワークを介して接続される、サーバ情報処理装置であって、前記第二のサーバ情報処理装置が管理するファイル（以下、異種ファイルと呼ぶ）へのアクセスを行うアクセス手段を備え、前記アクセス手段は、前記異種ファイルのディレクトリ構造の少なくとも一部に関する情報を検出して、該検出した情報に応じて、前記第一のファイルサーバプログラムが管理することができる形態を備えた、前記異種ファイルのディレクトリ構造のコピーを作成し、該作成した異種ファイルのディレクトリ構造のコピーを、前記第一のファイルサーバプログラムが管理するファイルのディレクトリ構造中に追加すると共に、前記情報の検出を周期的に行い、その度に異種ファイルのディレクトリ構造のコピーを更新するディレクトリ同期手段を備えることを特徴とするサーバ情報処理装置。

【請求項25】ファイルの共用を可能とする第一のファイルサーバプログラムを具備し、前記第一のファイルサーバプログラムとは共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置と、前記第一のファイルサーバプログラムを介して共用ファイルへアクセスするクライアントプログラムを具備する少なくとも1台のクライアント情報処理装置とが、ネットワークを介してそれぞれ接続される、サーバ情報処理装置であって、前記第二のサーバ情報処理装置が管理するファイル（以下、異種ファイルと呼ぶ）へのアクセスを行うアクセス手段を備え、前記アクセス手段は、前記各クライアントプログラムから送られてくる、前記異種ファイルへのアクセスをフックするフック手段と、前記アクセスが前記異種ファイルへのオープン要求であった場合に、該当する異

種ファイルのデータを、前記第一のファイルサーバプログラムが管理するディレクトリ構造内へコピーするファイルコピー手段と、前記ファイルコピー手段でコピーしたファイルを、コピー元へ書き戻すコピーバック手段と、前記アクセスが前記異種ファイルへのクローズ要求であった場合に、前記コピーバック手段を別プロセスとして起動するコピーバック起動手段とを備えることを特徴とするサーバ情報処理装置。

【請求項 26】請求項 2 記載の情報処理システムにおいて、前記サービス手段は、前記ディレクトリ同期手段から送られてくる更新すべきファイルの検索要求を受け入れ、前記コピーの対象となっている、前記ディレクトリ構造の少なくとも一部に含まれる異種ファイルのうち、前記前回の同期時刻以降に変更のあったファイルを検索し、変更があったファイルについての情報を前記ディレクトリ同期手段へ戻す、ディレクトリ同期サービス手段を備えることを特徴とする情報処理システム。

【請求項 27】請求項 7 記載の情報処理システムにおいて、前記サービス手段は、前記第一のサーバ情報処理装置から送られてくる、前記異種ファイルへのアクセスを開始するためのログイン要求を受け入れて、該受け入れたログイン要求毎に、該当する異種ファイルに対するアクセス要求を処理する、ユーザプロセスサービス手段を起動する、リモートログインサービス手段を備えることを特徴とする情報処理システム。

【請求項 28】請求項 27 記載の情報処理システムにおいて、前記ユーザプロセスサービス手段は、前記第一のサーバ情報処理装置によってリダイレクトされた、前記クライアントプログラムからのアクセス要求を受け入れ、該アクセス要求に該当する処理を該当する異種ファイルについて実行し、該実行結果を前記第一のサーバ情報処理装置へ送信することを特徴とする情報処理システム。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、情報処理装置のファイルアクセス手段及びその方法に係り、特に、ネットワークを介して、複数の異種ファイルシステム上の共用ファイルをアクセスする場合に好適な、異種ファイルアクセス手段を備えた情報処理システムおよびその方法に関する。

【0002】

【従来の技術】パーソナルコンピュータなどの情報処理装置が普及するにつれて、従来は各情報処理装置毎の磁気記憶装置上に保管していたユーザーのプログラムファイル、データファイルなどを一箇所の大容量磁気記憶装置に保管し、各情報処理装置のユーザーがデータを共用する、といった使い方が行われるようになってきた。

【0003】一般的には、ファイルサーバプログラムが搭載されたサーバ情報処理装置と、該サーバ情報処理装

置へアクセスするためのクライアントプログラムが搭載された複数のクライアント情報処理装置とをネットワークで接続し、サーバ情報処理装置上のファイルを各クライアント情報処理装置からアクセスし利用する。

【0004】また、サーバ情報処理装置上のファイルのセキュリティを保つため、ファイルサーバプログラムにはユーザー認証機構と、該ユーザー認証に基づいた、各ファイルへのアクセス権限設定／確認機構が具備されるのが一般的である。

【0005】上述のようなファイル共用システムの発達とネットワークの広域化を背景に、一台のクライアント情報処理装置から複数のサーバ情報処理装置上のファイルへのアクセスを行いたいというニーズが高くなっている。

【0006】ところが、複数のサーバ情報処理装置のファイルサーバプログラムが異なる種類であった場合、クライアント情報処理装置には、各ファイルサーバプログラムの種類に対応したクライアントプログラムを搭載しなければならない。このため、異なるクライアントプログラムが共存できない、動作に必要なメモリなどのハードウェア規模が大きくなる、あるいは、アクセス先によって操作方法が異なるといった問題等が生じる場合がある。

【0007】これらの問題を解決する従来技術としては、第二のサーバ情報処理装置上のファイルを第一のサーバ情報処理装置へと転送するファイル操作プログラムを第一のサーバ情報処理装置上に設け、また前記転送を指示するファイルアクセス要求プログラムをクライアント情報処理装置上に設ける方法が、特願平 5-152303 号によって開示されている。

【0008】この従来技術の方法では、アプリケーションなどからのファイル利用に先立って、まず、ユーザーが前記ファイルアクセス要求プログラムを用いて、第一のサーバ情報処理装置上のファイル操作プログラムに、第二のサーバ情報処理装置上から第一のサーバ情報処理装置上へのファイル転送を実行させる。その後、前記ファイルを第一のサーバ情報処理装置上のファイルとしてアクセスする。前記アクセスによってファイルの内容を変更した場合には、再びファイルアクセス要求プログラムを用いて、第一のサーバ情報処理装置上のファイル操作プログラムに、第一のサーバ情報処理装置上から第二のサーバ情報処理装置上へのファイル転送を実行させる。これによって、第二のサーバ情報処理装置上のファイルを直接アクセスした場合と同じ結果を得ることができる。したがって、複数の異なるクライアントプログラムをクライアント情報処理装置に搭載する必要がないという効果が達成される。

【0009】

【発明が解決しようとする課題】しかし、上記の従来例では、上述したような追加的な処理を必要とするため、

10

20

30

40

50

第二のサーバ情報処理装置上にあるファイルに対する操作性は、第一のサーバ情報処理装置上に元からあるファイルに対する操作性に比較して低いものとなっていた。

【0010】より具体的には、上記従来例では、第一のサーバ情報処理装置上には第二のサーバ情報処理装置上のファイルのディレクトリ構造に関する情報が存在しないため、第一のファイルサーバプログラム用に作成されたファイル一覧表示プログラムなどで第二のファイルサーバ上のファイル一覧を表示できないという課題があった。

【0011】また、第一のサーバ情報処理装置から第二のサーバ情報処理装置へのファイル書き戻しが終了するまで、次の処理が開始できないという課題があった。

【0012】また、複数のユーザーが同時に同一ファイルを利用した場合、不必要なファイル転送が発生したり、ユーザーが意図しないファイルの上書きが行われたりするという課題があった。

【0013】本発明は、上記のような課題を解決するためになされたもので、異なるファイルサーバプログラムを備えた複数のサーバ情報処理装置上のファイルに対してアクセスする場合でも、単一のファイルサーバプログラムを備えたサーバ情報処理装置上のファイルへアクセスする場合とほぼ同様な操作性を確保することができる、異種ファイルへのアクセスを可能とする情報処理システム及びその制御方法を提供することを目的とする。

【0014】より具体的には、本発明の他の目的は、第一のサーバ情報処理装置用のファイル一覧表示プログラムで、第二のサーバ情報処理装置上のファイル一覧を表示可能な情報処理システムおよびその制御方法を提供することにある。

【0015】また、本発明の他の目的は、第一のサーバ情報処理装置上に第二のサーバ情報処理装置上のディレクトリ構造を再現した場合にも、第一のサーバ情報処理装置のファイル記憶装置の容量消費を少なくすることができる、情報処理システム及びその制御方法を提供することにある。

【0016】また、本発明の他の目的は、第一のサーバ情報処理装置上に第二のサーバ情報処理装置上のディレクトリ構造を再現する際に第二のサーバ情報処理装置から第一のサーバ情報処理装置に送る情報量を少なくすることができる、情報処理システム及びその制御方法を提供することにある。

【0017】また、本発明の他の目的は、クライアントプログラムが、第二のサーバ情報処理装置上のファイルへのアクセスを終了し、該ファイルをクローズした後、第一のサーバ情報処理装置から第二のサーバ情報処理装置への書き戻し完了を待つことなく次の処理を実行可能な、情報処理システム及びその制御方法を提供することにある。

【0018】また、本発明の他の目的は、複数のユーザ

ーが、第二のサーバ情報処理装置上の同一ファイルをほぼ同時にアクセスする場合にも、第二のサーバ情報処理装置から第一のサーバ情報処理装置へのファイルコピーを一回で済ませることができる、情報処理システム及びその制御方法を提供することにある。

【0019】また、本発明の他の目的は、第二のサーバ情報処理装置上の同一ファイルへのアクセスが連続した場合に、第二のサーバ情報処理装置から第一のサーバ情報処理装置へのファイルコピーを一回で済ませることができる、情報処理システム及びその制御方法を提供することにある。

【0020】また、本発明の他の目的は、複数のユーザーが、第二のサーバ情報処理装置上の同一ファイルを同時にアクセスし内容を変更した場合にも、第一のサーバ情報処理装置から第二のサーバ情報処理装置へのファイルの書き戻しを一回で済ませることができる、情報処理システム及びその制御方法を提供することにある。

【0021】また、本発明の他の目的は、第二のサーバ情報処理装置上の同一ファイルへのアクセス直後に前記ファイルへの削除が発生した場合に、第一のサーバ情報処理装置から第二のサーバ情報処理装置へのファイルの書き戻しを行わないことができる、情報処理システム及びその制御方法を提供することにある。

【0022】

【課題を解決するための手段】上記目的は、ファイルの共用を可能とする第一のファイルサーバプログラムを具備する第一のサーバ情報処理装置と、前記第一のファイルサーバプログラムとは、共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置と、前記第一及び第二のサーバ情報処理装置を接続するネットワークとを備える情報処理システムであって、前記第一のサーバ情報処理装置は、前記第二のサーバ情報処理装置が管理するファイル（以下、異種ファイルと呼ぶ）へのアクセス処理を実行するアクセス手段を備え、前記第二のサーバ情報処理装置は、前記アクセス手段から送られてくる処理要求を受け入れて、該処理要求に対応する処理を実行するサービス手段を備え、前記アクセス手段は、前記サービス手段を介して前記異種ファイルのディレクトリ構造の少なくとも一部の構造に関する情報を検出して、該検出した情報に応じて、前記第一のファイルサーバプログラムが管理することができる形態を備えた、前記異種ファイルのディレクトリ構造のコピーを作成し、該作成した異種ファイルのディレクトリ構造のコピーを、前記第一のファイルサーバプログラムが管理するファイルのディレクトリ構造中に追加すると共に、前記情報の検出を所定のタイミングで実行し、その度に異種ファイルのディレクトリ構造のコピーを更新するディレクトリ同期手段を備えることを特徴とする情報処理システムにより達成される。

【0023】より具体的には、上記他の目的は、前記情報処理システムにおいて、前記情報の検出を例えば周期的に実行することで、第二のサーバ情報処理装置上のディレクトリ構造を第一のサーバ情報処理装置上に定期的にコピーすることによって達成される。

【0024】また、上記他の目的は、前記情報処理システムにおいて、前回ディレクトリ同期を実行した同期時間を記憶する記憶手段と、前回のディレクトリ同期を実行した時間以降に更新された異種ファイルを検索し、前記検索手段が示す更新ファイルの情報だけを第二のサーバ情報処理装置から第一のサーバ情報処理装置に送る検索手段と、前記更新ファイルの情報にしたがい、第一のサーバ情報処理装置のディレクトリ構造中に追加されている、異種ファイルのディレクトリ構造のコピーを更新することによって達成される。

【0025】また、上記他の目的は、前記情報処理システムにおいて、前記ディレクトリ同期手段が、第一のサーバ情報処理装置上に通常ファイルのコピーを作成する場合に、そのファイルに格納するデータをコピーせず、例えば中身の無いスパースファイルとすることによって達成される。

【0026】上記目的は、また、ファイルの共用を可能とする第一のファイルサーバプログラムを具備する第一のサーバ情報処理装置と、前記第一のファイルサーバプログラムとは、共用ファイルへのアクセスインタフェースが異なる第二のファイルサーバプログラムを具備する第二のサーバ情報処理装置と、前記第一のファイルサーバプログラムを介して前記第一のサーバ情報処理装置上の共用ファイルへアクセスするクライアントプログラムを具備する少なくとも1台のクライアント情報処理装置と、前記第一のサーバ情報処理装置及び前記少なくとも1台のクライアント情報処理装置を接続する第一のネットワーク、及び、前記第一及び第二のサーバ情報処理装置を接続する第二のネットワークとを備える情報処理システムであって、前記第一のサーバ情報処理装置は、前記第二のサーバ情報処理装置が管理する異種ファイルへのアクセスを行うアクセス手段を備え、前記第二のサーバ情報処理装置は、前記アクセス手段と対応して動作するサービス手段を備え、前記アクセス手段は、前記各クライアントプログラムから送られてくるファイルへの処理要求を受け入れ、該処理要求が前記第二のファイルサーバプログラムが管理するファイル（以下、異種ファイルと呼ぶ）への処理要求を含むかどうかを判断し、それを含むと判断された場合には、前記処理要求に含まれている異種ファイルへの処理要求を、前記第二のファイルサーバプログラムで利用できる第二の処理要求に変換し、該変換した第二の処理要求を前記第二のサーバ情報処理装置へ送るものであり、前記サービス手段は、前記第一のサーバ情報処理装置から送られてくる、前記第二の処理要求を受け入れ、それに対応して予め定められて

いる処理を実行することを特徴とする情報処理システムにより達成される。

【0027】また、上記他の目的は、前記情報処理システムにおいて、前記アクセス手段が、各クライアントプログラムから送られてくる異種ファイルへのアクセスをフックするフック手段と、前記アクセスが前記異種ファイルへのオープン要求であった場合に該当するファイルのデータを前記サービス手段を介して前記第一のファイルサーバプログラムが管理するディレクトリ構造内へコピーするファイルコピー手段と、前記ファイルコピー手段でコピーしたファイルを前記サービス手段を介して前記第二のサーバ情報処理装置に書き戻すコピーバック手段と、前記アクセスが前記異種ファイルへのクローズ要求であった場合に前記コピーバック手段を別プロセスとして起動するコピーバック起動手段とを備える。

【0028】また、上記他の目的は、前記情報処理システムにおいて、前記コピーしたファイルの名称、リードオープン、ライトオープンしている数、コピーバック待ちフラグ、オープン待ちフラグ、削除待ちフラグ等を記憶し、その時点での状態に応じて、コピーバック処理の実行を制御する。

【0029】上記目的は、また、情報処理システムの制御方法において、前記異種ファイルのディレクトリ構造の少なくとも一部の構造に関する情報を検出して、該検出した情報に応じて、前記第一のファイルサーバプログラムが管理することができる形態を備えた、前記第二のサーバ情報処理装置が管理するファイルのディレクトリ構造のコピーを作成し、該作成したディレクトリ構造のコピーを、前記第一のファイルサーバプログラムが管理するファイルのディレクトリ構造中に追加する処理と、前記情報の検出を所定のタイミングで実行し、その度に前記異種ファイルのディレクトリ構造のコピーを更新する処理とを含むことを特徴とする情報処理システムの制御方法により達成される。

【0030】上記目的は、また、情報処理システムの制御方法において、前記クライアントプログラムから送られてくる、前記第二のサーバ情報処理装置が管理するファイル（以下、異種ファイルと呼ぶ）へのアクセスをフックする処理と、前記アクセスが前記異種ファイルへのオープン要求であった場合に、該当するファイルのデータを、前記第一のファイルサーバプログラムが管理するディレクトリ構造内へコピーする処理と、前記ファイルコピー手段でコピーしたファイルを、前記サービス手段を介してコピー元に書き戻す処理と、前記アクセスが前記異種ファイルへのクローズ要求であった場合に、前記コピーバック手段を別プロセスとして起動する処理とを含むことを特徴とする情報処理システムの制御方法により達成される。

【0031】

【作用】本発明を適用した情報処理システム及びその制

御方法によれば、第一のファイルサーバプログラムは、元来自身が管理するファイルと共に、第二のファイルサーバプログラムが管理する異種ファイルのコピーしたファイルの管理、あるいは、アクセス要求に対応する処理を実行することができる。

【0032】よって、ユーザは、第一のサーバファイルプログラムを通して、当該プログラムが管理するファイルだけでなく、第二のファイルサーバプログラムが管理する異種ファイルへもアクセスすることができる。

【0033】具体的には、ディレクトリ同期手段は、第二のサーバ情報処理装置上のディレクトリ構造の少なくとも一部を、第一のサーバ情報処理装置上のディレクトリ構造中に定期的にコピーするので、第一の情報処理装置上のファイル一覧を表示すれば、第二の情報処理装置上のファイル一覧に関する情報も同時に得ることができる。

【0034】また、ディレクトリ同期手段は、第二のサーバ情報処理装置のディレクトリ構造を第一のサーバ情報処理装置上に定期的にコピーする際に実行する前記第一のサーバ情報処理装置上へのファイル作成において、ディレクトリファイル以外の通常のファイルを、中身の

ないスパーズファイルとして作成する。

【0035】これによって、前記ファイルが使用する第一の情報処理装置の磁気ディスク中の記憶容量は、ファイル一覧表示中にファイルサイズとして示されているものとは無関係な、非常に小さなサイズとなる。

【0036】また、検索手段は、ファイルに変更があったかどうかを、例えば記憶手段に記憶されているディレクトリ同期を実行した前回の同期時間と第二のサーバ情報処理装置上のファイルの最終更新日付とを比較することで検出し、前回のディレクトリ同期を実行した時間以降に更新されたファイルの情報だけを、第一のサーバ情報処理装置に送らせる。更新手段は、前記更新されたファイルについての情報にしたがい、第一のサーバ情報処理装置のディレクトリ構造中に追加されている、前記異種ファイルのディレクトリ構造のコピーを更新する。

【0037】このように前回のディレクトリ同期実行時刻以後に変更されたファイルだけを同期処理の対象とすることによって、第二のサーバ情報処理装置から第一のサーバ情報処理装置に送る情報量を少なくすることができる。

【0038】また、フック手段は、第一のサーバ情報処理装置のファイルシステムに対するアクセス要求をフックし、該アクセス要求が、クライアント情報処理装置から前記ディレクトリ同期手段によってコピーされ作成された異種ファイルのディレクトリ構造のコピー内に属するファイルへのオープン、クローズ、削除要求のいずれかかどうかを判定する。

【0039】前記判定の結果、オープン要求であった場合には、ファイルコピー手段によって該ファイルの実体

である、当該ファイルに格納されているデータを、第二のサーバ情報処理装置から第一のサーバ情報処理装置へとコピーする。このとき、該ファイルに既にデータが存在するかを調べ、存在しない場合にだけコピーを行う。また、前記オープン要求がリードモードかライトモードかによって、該ファイルについてのリードオープン、ライトオープン数を設定する。

【0040】また、上記フック手段での判定結果が削除要求であった場合には、該ファイルが存在するかどうかを調べ、さらにコピーバック待ちフラグを調べセットされていたら、削除待ちフラグをセットする。

【0041】また、上記フック手段での判定結果がクローズ要求であった場合には、コピーファイルテーブル手段のリードオープン、ライトオープン数を該クローズに対応するオープンで加算していた分だけ減算する。この結果、ライトオープン数が0となれば、コピーバック手段を別プロセスとして起動し、前記フック処理自体は終了する。

【0042】コピーバック手段は、コピーバック待ちフラグを設定し、その後オープン待ちフラグ、削除待ちフラグが設定されていないかを監視しながら、予め定めた時間だけ待つ。前記待ち時間の間に削除待ちフラグがセットされれば、以後のコピーバック処理を中止する。また、オープン待ちフラグがセットされれば、待ち時間の完了を待たずに第一のサーバ情報処理装置にコピーしたファイルを再び第二のサーバ情報処理装置に書き戻すコピーバック処理を行う。

【0043】コピーバック処理が終了し、オープン待ちフラグがセットされず、かつリードオープン数、ライトオープン数が共に0の場合には、該ファイルのコピーは不要であるので、該ファイルのコピーを中身のないスパーズファイルとする。

【0044】以上の構成の処理によれば、複数ユーザが同時に同一ファイルを使用する際にも無駄なコピー処理が発生しない。さらに、複数のユーザが使用中であったり、クローズに続いて再度オープンされる場合にスパーズファイル化せず、コピーファイルを引き続き利用できる。さらに、クローズに続いて削除される場合に、不要なコピーバックをせずに済む。

【0045】また、コピーバック手段は、別プロセスとしてバックグラウンドで動作するので、前記コピーバック手段の完了を待つことなくクライアント情報処理装置からのクローズ要求は終了するので、ユーザは直ちに次の処理を行うことができる。

【0046】

【実施例】以下、本発明の一実施例を図を用いて説明する。

【0047】最初、本実施例の情報処理システム全体の構成を説明する。本実施例のシステムは、図1に示すように、第一のクライアント情報処理装置110、第二の

クライアント情報処理装置140、第一のサーバ情報処理装置120、及び、第二のサーバ情報処理装置130を有する。

【0048】クライアント情報処理装置110、140、サーバ情報処理装置120、130には、中央処理装置（CPU）1a、1d、1b、1c、メモリ2a、2d、2b、2c、LANコントローラ3a、3d、3b、3c、ディスクコントローラ4a、4d、4b、4c、磁気ディスク5a、5d、5b、5cがそれぞれ具備されている。各情報処理装置はローカルエリアネットワーク（LAN）101を介して相互に接続されており、各々のLANコントローラ3を介して通信ができる構成を有している。

【0049】なお、本実施例ではネットワークとしてLANを用いているが、本発明においては、各情報処理装置を互いに接続する通信路の具体的な形態について限定するものではなく、例えばワイドエリアネットワーク（WAN）を使用する構成としても良い。

【0050】クライアント情報処理装置110のメモリ2a上には、オペレーティングシステムプログラム（OS）6a、ファイルサーバプログラム9b用のクライアントプログラム7a、アプリケーションプログラム8aが、起動時に磁気ディスク5aからロードされる。クライアント情報処理装置140のメモリ2d上には、オペレーティングシステムプログラム（OS）6d、ファイルサーバプログラム9c用のクライアントプログラム7d、アプリケーションプログラム8dが、起動時に磁気ディスク5dからロードされる。

【0051】第一のサーバ情報処理装置120のメモリ2b上には、ゲートウェイプログラム20、OS6b、第一のファイルサーバプログラム9bが起動時に磁気ディスク5bからロードされる。また、第二のサーバ情報処理装置130のメモリ2c上には、エージェントプログラム30、OS6c、第二のファイルサーバプログラム9cが起動時に磁気ディスク5cからロードされる。

【0052】これらサーバにロードされるプログラムのうち、ゲートウェイプログラム20、エージェントプログラム30が、本発明で新たに追加した特徴的な構成であり、その他の構成は従来のファイルサーバシステムで用いられていたものである。

【0053】なお、本実施例では、ゲートウェイプログラム20が第一のサーバ情報処理装置120に、エージェントプログラム30が第二のサーバ情報処理装置130に記憶されている場合の構成を例にとって説明するが、各サーバ情報処理装置がそれぞれ、他方へのアクセス及び他方からのアクセスを可能とする、ゲートウェイプログラム及びエージェントプログラムを記憶する構成としても良い。

【0054】クライアント情報処理装置110のユーザー（以下、単にユーザーと呼ぶ）は、クライアントプロ

グラム7aを介して、サーバ情報処理装置120のファイルサーバプログラム9bにログインし、アプリケーションプログラム8aから磁気ディスク5b上の共用ファイルをアクセスする。

【0055】ここで、本実施例のゲートウェイプログラム20とエージェントプログラム30とにより、ユーザーが前記アクセスと全く同じ方法で、磁気ディスク5b上のファイルであるがごとく、磁気ディスク5c上の共用ファイルへアクセスできる仕組みが提供される。

【0056】ゲートウェイプログラム20は、初期化、終了処理を行うメインルーチン21、ユーザーが第一のファイルサーバプログラム9bへログインしたことを検出して第二のファイルサーバプログラム9cへの自動ログインを指示するログインルーチン22、前記自動ログインを各ユーザー対応に実行するユーザープロセスルーチン23、前記ユーザーからのファイルアクセスをフックし、第二のサーバ情報処理装置130へのアクセスへとリダイレクトするファイルアクセスフックルーチン24、第二のサーバ情報処理装置130から第一のサーバ情報処理装置120上にコピーされ修正を加えられたファイルを、再び第二のサーバ情報処理装置130へ転送するコピーバックルーチン25、及び、第二のサーバ情報処理装置130上のディレクトリ構造の指定した部分と同じ構造を第一のサーバ情報処理装置120上に作成するディレクトリ同期ルーチン26を具備する。

【0057】エージェントプログラム30は、前記ユーザープロセスルーチン23からのログイン要求をうけつけるリモートログインサービスルーチン31、前記ログイン受け付け時に起動され、第二のファイルサーバプログラム9cへのログインを実行し、その後、前記ファイルアクセスフックルーチン24によってリダイレクトされた各ユーザーからのファイルアクセスに応答するユーザープロセスサービスルーチン32、及び、前記ディレクトリ同期ルーチン26からの要求に応じて、第二のサーバ情報処理装置130のディレクトリ構造情報を返すディレクトリ同期サービスルーチン33を具備する。

【0058】次に、本システムの動作の概要を説明する。図21は、本発明のゲートウェイプログラム20を起動する前に第一のサーバ情報処理装置120の磁気ディスク5b上に作成されているファイルのディレクトリ構造の一例を示す図である。ここで、括弧〔 〕で囲んだファイル名のファイルは、ディレクトリファイルであり、その下につながっているファイルの情報が格納されている。図22は、第二のサーバ情報処理装置130の磁気ディスク5c上に作成されているファイルのディレクトリ構造の一例を示す図である。

【0059】本実施例においては、ゲートウェイプログラム20を起動すると、第一のサーバ情報処理装置120の磁気ディスク5b上に作成されるファイルのディレクトリ構造は、例えば図23に示すような構造となる。

ここで、破線で囲った部分には、上述のディレクトリ同期ルーチン26の機能によって、図22で示した第二のサーバ情報処理装置130上のディレクトリ構造の内、`research/hard`から下位の構造がコピーされている。

【0060】コピーしたディレクトリ構造内のファイルは、ファイルサーバプログラム9bがOS6bを介して管理する通常のファイルとして磁気ディスク5b上に作成する。これによって、クライアント情報処理装置110上のアプリケーションソフトウェア8aからは、第二のサーバ情報処理装置130上のディレクトリ構造の一部を、第一のサーバ情報処理装置120上のディレクトリ構造として参照することができる。

【0061】ここで、コピーしたディレクトリ構造内のファイルの内、ディレクトリファイルでない通常ファイルは、その中身のデータを第二のサーバ情報処理装置130からコピーせず、1バイトだけのダミーデータを格納したスパースファイルとして作成する。図25にスパースファイルの例を示す。ここで、ファイルVOL1/mpdl/circ/log1は、そのファイルサイズが、524950バイトであるが、データは先頭から24950バイト目に1バイトだけ書かれているだけである。このようなファイルでは、先頭のブロックと最終ブロックだけが実際に磁気ディスク5b上の記憶領域を使い、他のブロックは磁気ディスク5bの領域を使用しない。本例では、ブロックサイズが4096バイトであるので、ファイルサイズが524950バイトでも実際に使用する磁気ディスク5bの容量は4096の2倍の8192バイトで済む。

【0062】このように、ディレクトリ構造のコピー部分のファイルにデータの実体のないスパースファイルを用いることによって、ファイル情報としてのファイルサイズはコピー元のサイズを反映し、かつ磁気ディスク5bの使用容量を少なくすることができる。

【0063】クライアント情報処理装置110のアプリケーションプログラム8aからのファイルのデータへのアクセスは、以下のように実現される。

【0064】ゲートウェイプログラム20のファイルアクセスフックルーチン24が、アプリケーションプログラム8aからのファイルオープン要求を検出し、第二のサーバ情報処理装置130から該ファイルのデータを第一のサーバ情報処理装置120上のスパースファイル化している当該ファイルへ上書きコピーする。前記オープン後の該ファイルに対するアプリケーションプログラム8aからのリードライトアクセスは、ゲートウェイプログラム20の関与無しに行われる。一方、アプリケーションプログラム8aからのクローズ要求を検出すると、該ファイルへのデータの書き込みが行われたかどうかを判定し、書き込みが行われていたときには上記コピーバックルーチン25が、第二の情報処理装置130ヘデー

タを書き戻す。また、クローズ後は再び第一の情報処理装置上の当該ファイルを上述のスパースファイルとする。このコピーバック処理は、一端開始の指示が発行されるされると、ユーザーとの関与なしにバックグラウンドで行なわれる。このため、アプリケーションプログラム8aはコピーバック処理の終了を待つことなく、次の処理を行うことができる。また、コピーバック処理はファイルのクローズ後すぐには開始せず、しばらく時間をおいてから開始する。この時間をコピーバックディレイタイムと呼ぶ。

【0065】システム管理者が作成し、磁気ディスク5b上に格納するゲートウェイパラメータ設定ファイル199の一例を図24に示す。本図の一行目(2410)では、第二の情報処理装置130上のディレクトリ構造のどの部分を第一の情報処理装置120のどのディレクトリに割り当てるかを示している。ここでは、“SRV2”という名称の第二のサーバ情報処理装置130の“`research/hard`”以下のディレクトリ構造を、第一のサーバ情報処理装置120の“VOL1/mpdl”に割り当てる例を示している。二行目(2420)は、アプリケーションプログラム8aがファイルをクローズしてからコピーバックルーチン25が、ファイルのコピーバックを開始するまでの時間であるコピーバックディレイタイムを指定する。本例では、500m秒を指定する例を示している。三行目(2430)は、ディレクトリ同期を行う時間間隔を指定する。ここでは、6000秒、即ち10分とした例を示す。

【0066】以下、上述した各ルーチンの処理を詳細に説明する。

【0067】最初、ルーチンの説明に先立ち、ゲートウェイプログラム20内の各ルーチンでメモリ2b上に設定、参照するテーブルの例を示す。

【0068】図14は、マウントパステーブル1400の一例である。マウントパステーブル1400は、上述のゲートウェイパラメータ設定ファイル199に記述された、第二の情報処理装置130上のディレクトリ構造のどの部分を第一の情報処理装置120のどのディレクトリに割り当てるかの情報を格納するテーブルである。複数の割り当てを可能とするため、マウントパステーブル1400は複数のノード1490と、最初のノードを示す先頭ポインタ1401から構成される線形リスト構造としている。各ノード1490には、次のノードを指し示すポインタ1402、第一の情報処理装置120のディレクトリのパスであるマウントパス1403、及び、第二の情報処理装置130のディレクトリのパスであるリモートパス1404が格納される。最後のノードのポインタ1402は、空きを示す“NULL”が格納されている。

【0069】図15は、ユーザーコネクションテーブル1500の一例である。ユーザーコネクションテーブル

1500は、最初のノードを示す先頭ポインタ1501とログインした各ユーザーに対応した複数のノード1590とから構成される。各ノード1590は、次のノードを指し示すポインタ1502、第一のファイルサーバプログラム9bがログインしたユーザーに割り当てたコネクション番号1503、ログインルーチン22が各ユーザー毎に起動し割り当てたユーザープロセスのプロセス番号1504、該ユーザープロセスが、エージェントプログラム30のユーザーサービスルーチン32と通信するためのソケット番号1505、及び、各ユーザーが使用中の第二の情報処理装置130上のファイルに関する情報を記憶したファイルハンドルテーブル1600へのポインタであるファイルハンドルテーブルポインタ1506から構成される。

【0070】図16は、前記ファイルハンドルテーブル1600の一例である。このテーブルはユーザー毎に作成され、テーブルの各ノード1690は、次のノードを指し示すポインタ1602、第一の情報処理装置120上のパス名を含むファイル名1603、該ファイルをオープンした際にOS6bから与えられるのファイルハンドル番号1604、第二の情報処理装置130上のパス名を含むリモートファイル名1605、及び、該ファイルをオープンした際にOS6cから与えられるのリモートファイルハンドル番号1606から構成される。

【0071】図17は、第二の情報処理装置から第一の情報処理装置へファイルのデータが一時的にコピーされているファイルの情報を示すコピーファイルテーブル1700の一例である。コピーファイルテーブル1700は、最初のノードを示す先頭ポインタ1701と前記ファイル毎に対応した複数のノード1790とから構成される。各ノード1790は、次のノードを指し示すポインタ1702、第一の情報処理装置120上のパス名を含むファイル名1703、該ファイルに対して第一の情報処理装置120上でリードモードでオープンしている数を示すリードオープン数1704、同じくライトオープン数1705、該ファイルが第二の情報処理装置130へ書き戻し処理が起動されていることを示すコピーバックフラグ1706、該ファイルに対してオープン待ちのプロセスがあることを示すオープン待ちフラグ1707、及び、同じく削除待ちのプロセスがあることを示すイレース待ちフラグ1708から構成される。

【0072】以下、上述の各ルーチンの処理フローを説明する。

【0073】最初に、ゲートウェイプログラム20のメインルーチン21の処理フローを、図2を参照して説明する。

【0074】本処理フローにおいて、ステップ201では、上記ユーザーコネクションテーブル1500の先頭ポインタ1501、ユーザーコネクションテーブル1500の先頭ポインタ1501などのシステム変数を初期

化する。ステップ203では、予めシステム管理者が作成し磁気ディスク5b上に格納しているゲートウェイパラメータ設定ファイル199を参照し、上記マウントパステーブル1400を作成する。

【0075】ステップ204でディレクトリ同期ルーチン26を新たなプロセスとして起動し、上記マウントパステーブルで設定した対応関係にしたがい、第二の情報処理装置130の磁気ディスク5c上のディレクトリ構造の一部を第一の情報処理装置120の磁気ディスク5b上に再現する。なお、各プロセスは、OSのタイムスライス機能によって、並行処理が行われる。

【0076】ステップ205から207では、ファイルサーバプログラム9b等から発行されるファイルアクセス要求をフックするため、OS6bのファイルシステムへのファンクションコールのサービスベクタを書き換える。ここでサービスベクタとは、該ファンクションコール発行時に実行されるCPU1bのソフトウェア割り込み命令で参照される分岐先アドレスのことである。まず、ステップ205でOS6bが設定したファンクションコールのサービスベクタを取得し、ステップ206で前記サービスベクタをシステム変数として記憶し保存する。ステップ207では、本ゲートウェイプログラム20内のファイルアクセスフックルーチン24のエントリーアドレスを新たなファイルアクセスサービスベクタとして設定し、ファイルアクセス要求をフックするための設定を終える。

【0077】次に、ステップ208でログインルーチン22を新たなプロセスとして起動し、ユーザーからのログイン監視を開始する。ステップ209では自らスリープ状態となり停止指令を待つ。この時点でゲートウェイプログラム20の初期化は終了し、ユーザーは第二の情報処理装置130上のファイルをアクセスできるようになる。

【0078】システム管理者がゲートウェイプログラム20の停止命令を発行すると、メインルーチンはスリープ状態から抜け、ステップ210以降の終了処理を行う。ステップ210では、ログインルーチン22を終了させるためのシグナルをログインプロセスに送り、ステップ211では同様にディレクトリ同期プロセスへの終了指示を送る。ステップ212では、ステップ207で書き換えたファイルアクセスサービスベクタをステップ206で保存しておいた元の値に戻す。以上でメインルーチン21を終了する。

【0079】次に、リモートログインルーチン22の処理フローを、図3を参照して説明する。本処理フローにおいて、ステップ301では、ファイルサーバプログラム9bへのユーザーのログインまたはログアウト、上述のメインルーチン21からのログインプロセス停止指示などのイベントを監視し待つ。該イベントが発生するとステップ302でログインかどうかを判定し、ログイン

の場合は、ステップ303でユーザープロセスルーチン23を新たなプロセスとして起動する。

【0080】一方、ステップ302でログインではないと判定した場合はステップ304を実行し、ログアウトかどうかを調べる。ログアウトだった場合にはステップ305を実行し、ログアウトでなければステップ306を実行する。前記ステップ305では、上述のユーザーコネクションテーブル1500を参照し、ログアウト要求元のコネクション番号1503からユーザープロセス番号1504を調べ、該プロセスにログアウト指示のシグナルを送る。

【0081】ステップ306では、発生したイベントがサービス停止指示かどうかを判定し、停止指示であった場合には、ログインプロセスを終了する。前記停止指示以外のイベントであった場合には、再びステップ301でのイベント待ちを行う。

【0082】次に、上記ログインルーチン22のステップ303で生成、起動されるユーザープロセスルーチン23の処理フローを、図4を参照して説明する。本処理フローでは、最初、ステップ400で、第二の情報処理装置130上のエージェントプログラム30のリモートログインサービスルーチン31に対してソケットを作成し新たな通信コネクションを確立する。次にステップ401で、前記リモートログインサービスルーチン31に対して、第二のファイルサーバプログラム9cへのログインを依頼する。

【0083】このとき、前記リモートログインサービスルーチン31は、まずユーザープロセスサービスルーチン32を新たなプロセスとして起動し、ここからログインを実行する。前記ユーザープロセスサービスルーチン32は、リモートログインサービスルーチン31から引き継いだソケットを用いて前記ユーザープロセスルーチン23に前記ログインの結果を伝える。ステップ402では該結果を判定し、ログイン成功であった場合にはステップ403以降を実行し、ログイン失敗であった場合には前記ユーザープロセスルーチンを終了するように制御する。

【0084】ステップ403では、前記ユーザーコネクションテーブル1500に新たなノード1590を追加し、コネクション番号1503、プロセス番号1504、ソケット番号1505、ファイルハンドルテーブルポインタ1506の各値を設定する。続くステップ404では、前記ログインルーチン22からログアウト指示を受け取るまでスリープして待つ。前記ログアウト指示を受け取ると、ステップ405で、第二のファイルサーバプログラム9cからのログアウトを、ユーザープロセスサービスルーチン32に依頼し実行する。前記ログアウトの後、ユーザープロセスルーチン23は終了する。

【0085】次に、ファイルアクセスフックルーチン24の処理フローを、図5を参照して説明する。上述のメ

インルーチン21で行ったファイルアクセスサービスベクタ書き換えによって、OS6bのファイルシステムへのファンクションコールが発行される度に本ルーチンが呼ばれる。

【0086】本ルーチンが呼ばれると、ステップ501でファイルアクセス要求元のコネクション番号が、ユーザーコネクションテーブル1500にあるかどうかを調査する。該調査の結果、ユーザーコネクションテーブル1500に存在しなければ、ファイルアクセスをリダイレクトする対象の要求元（ユーザー）ではないので、ステップ560で元のベクタをコールし本来の処理を行った後、ステップ599で本ルーチンの呼び出し元へリターンする。

【0087】ステップ502では、ファイルアクセス要求のあったファイル（以下単に要求ファイルと呼ぶ）が、第二の情報処理装置130のディレクトリをマウントしたディレクトリ部分に含まれるかどうかを調査する。該調査は、マウントパステーブル1400を参照し、マウントパス1403と要求ファイルの完全なパス名とを比較することによって行う。

【0088】なお、ファイルアクセス要求がクローズの場合には、ファイル名ではなくファイルハンドル番号が引き渡されるが、この場合には、まず、ユーザーコネクションテーブル1500のファイルハンドルテーブルポインタ1506から当該ユーザーのファイルハンドルテーブル1600の先頭ポインタを求め、次に前記ポインタが示すファイルハンドルテーブル1600内のファイルハンドル番号1604に該当するものがあるかを調査する。前記調査の結果、対象のファイルでなければ、ステップ560で元のベクタをコールし本来の処理を行った後、ステップ599で本ルーチンの呼び出し元へリターンする。

【0089】次にステップ505、510、520、530、540で、ファイルオープン、ファイルクローズ、ファイルイレース、ディレクトリ作成、ディレクトリ削除のいずれかの要求かどうかをそれぞれ判定する。

【0090】ファイルオープンの場合には、まず、ステップ506でオープンプリフックルーチン50をコールする。ステップ507では、該コールの結果がエラーであったかどうかを調べ、エラーであった場合には続くステップ508をスキップする。ステップ508では、保存してある元のファイルアクセスサービスベクタを参照しそのアドレスをコールすることによって、OS6bのファイルシステムにオープン処理を要求する（以下、このような処理を単に元のベクタをコールするという）。次にステップ509でオープンポストフックルーチン51をコールし、ステップ599でファイルオープンの呼び出し元にリターンする。

【0091】また、ファイルクローズの場合には、まず、ステップ511でクローズプリフックルーチン53

をコールする。ステップ512では、該コールの結果がエラーであったかどうかを調べ、エラーであった場合には続くステップ513をスキップする。ステップ513では、もとのベクタをコールし、ステップ599でファイルクローズの呼び出し元にリターンする。

【0092】ファイルイレース、ディレクトリ作成、ディレクトリ削除の場合も、図5に示すようにファイルクローズの場合と同様の処理を行う。

【0093】以上述べたように、本実施例のファイルアクセスフックルーチン24の処理によって、OS6bに
10 に対するファイルアクセス要求をフックし、第二の情報処理装置のファイルアクセスへとリダイレクトするための各処理ルーチンを呼び出すことができる。

【0094】次に、オープンプリフックルーチン50の処理フローの一例を、図6を参照して説明する。本ルーチンは、上記図5のファイルアクセスフックルーチン24のステップ506でコールされるものである。

【0095】本処理フローにおいては、最初、ステップ603で、ファイルハンドルテーブル1600に新たな
20 ノード1690を追加する。ステップ605では、コピーファイルテーブル1700の各ノード1790のファイル名1703に前記要求ファイル名が存在するかどうかを調べ、存在すればさらにそのノード1790のコピーバックフラグ1706を参照する。該参照の結果コピーバック中ならば、ステップ606で当該ノード1790のオープン待ちフラグ1707をセットし、さらにステップ607で当該ノードのコピーバックフラグ1706を監視し、コピーバック終了を待つ。コピーバックが終了すれば、ステップ608で前記オープン待ちフラグ
30 1707を解除する。ステップ605での判定結果がコピーバック中でなければ、上記ステップ606、607、608は実行されない。

【0096】ステップ610では、要求ファイルのオープンモードを第二のファイルサーバプログラム9cが受け付け可能かつ要求元のオープンモードになるべく等価なオープンモードに変換する。ここで、オープンモードとは、ファイルをリードモードでオープンするかライトモードでオープンするか、ファイルが存在しなければ作成するかどうか、排他制御はどうするかといった指定を行うものである。

【0097】ステップ612では、エージェントプログラム30のユーザープロセスサービスルーチン32に対し、要求ファイルに対応する第二の情報処理装置130上のファイルのオープンを依頼する。このとき、上記変換後のオープンモードを用いる。前記オープンの結果を前記ユーザープロセスサービスルーチン32から受け取ると、ステップ613で該オープンが成功したかどうかを判定し、失敗していた場合にはステップ660を実行する。

【0098】ステップ660では、失敗の原因を調べ、

オープンファイル数が規定値を越えていた為であった場合にはステップ661で時間待ちをした後再びステップ612に戻りリトライする。他の原因であった場合には、ステップ665でエラーコードを設定してステップ699で呼び出し元へリターンする。このリトライ処理は、第一のOS6bと第二のOS6cで許容されている同時オープンファイル数が異なることや、複数のコピーバックルーチン25がマルチプロセスで実行されるために、ファイルサーバプログラム9bを介してオープン
10 されているファイル数よりも多くのファイルが同時にオープンされるために必要となる処理である。上記ステップ613で、オープンが成功していた場合は、まずステップ615でファイルハンドルテーブル1600の当該ノード1690のリモートファイルハンドル番号1605に、前記オープンでユーザープロセスサービスルーチン32から返されるファイルハンドル番号を格納する。つぎにステップ620でコピーファイルテーブル1700を検索し、要求ファイルのデータが既に第二の情報処理装置130から第一の情報処理装置120上へとコピー
20 されているかを調べる。コピー済みであった場合にはステップ650を、コピー済みでなかった場合にはステップ622を実行する。

【0099】ステップ622からステップ630では、要求ファイルのデータを第二の情報処理装置130から第一の情報処理装置120上へとコピーする。まず、ステップ622で、第一の情報処理装置120のOS6bに対して要求ファイルのオープンを指示する。ファイルがなければ新たに作成する。次にステップ625で、第二の情報処理装置130から当該ファイルのデータを読み出す。この読出しは、上記ユーザープロセスサービス
30 ルーチン32を介して行う。前記読み出したデータは、ステップ626で、第一の情報処理装置130上のスパーズファイル化されている前記要求ファイルに書き込まれる。ステップ627では、全内容のコピーを終了したかどうか判定し、終了していなければステップ625以下を繰り返し実行する。ステップ630では、第一の情報処理装置120のOS6bに対して要求ファイルのクローズを指示する。

【0100】ステップ631では、コピーファイルテーブル1700に、新たなノード1790を追加し、第一
40 の情報処理装置120上でのファイル名1703を設定する。ステップ650では、要求ファイルのオープンモードにしたがって、前記ノード1790のリードオープン数、ライトオープン数の一方または両方に1を加える。以上述べた処理の後、ステップ699でオープンプリフックルーチンを終了し、呼び出し元へリターンする。

【0101】次に、オープンポストフックルーチン51の処理フローの一例を図7を参照して説明する。ステップ710で、本ルーチンが呼ばれる前に元のベクタをコ

ールすることによって実行したOS 6 bに対するファイルオープンが正常終了したかどうかを調べ、正常終了していた場合はステップ7 2 0を実行する。このステップ7 2 0では、ファイルハンドルテーブル1 6 0 0の該当するノード1 6 9 0のローカルファイルハンドル番号1 6 0 4に、前記オープンでOS 6 bから返されるファイルハンドル番号を格納する。

【0 1 0 2】一方、OS 6 bに対するファイルオープンが異常終了していた場合には、該オープンを無効とし、各テーブルへの登録情報を削除するためにステップ7 3 0以下を実行する。ステップ7 3 0では、オープンプリフックルーチン5 0で追加したファイルハンドルテーブル1 6 0 0の当該ノード1 6 9 0を削除する。続くステップ7 3 1では、同じくオープンプリフックルーチン5 0でカウントアップしたコピーファイルテーブル1 7 0 0のリードオープン数1 7 0 4、ライトオープン数1 7 0 5を前記カウントアップ前の値に戻す。

【0 1 0 3】以上の処理の後、ステップ7 9 9で呼び出し元にリターンし、オープンポストフックルーチン5 1を終了する。

【0 1 0 4】次に、イレースプリフックルーチン5 2の処理フローの一例を、図8を参照して説明する。まず、ステップ8 1 0で、コピーファイルテーブル1 7 0 0の各ノード1 7 9 0のファイル名1 7 0 3に要求ファイル名が存在するかどうかを調べ、存在すればさらにそのノード1 7 9 0のコピーバックフラグ1 7 0 6を参照する。該参照の結果コピーバック中ならば、ステップ8 2 0で当該ノード1 7 9 0のイレース待ちフラグ1 7 0 7をセットし、さらにステップ8 3 0で当該ノードのコピーバックフラグを監視し、コピーバック終了を待つ。

【0 1 0 5】コピーバック中でなかった場合、あるいはコピーバックが終了するとステップ8 4 0が実行される。ステップ8 4 0では、エージェントプログラム3 0のユーザープロセスサービスルーチン3 2に対し、要求ファイルに対応する第二の情報処理装置1 3 0上のファイルの削除を依頼する。ステップ8 5 0では、前記削除が正常終了したかどうかを判定し、異常終了の場合にはステップ8 6 0でエラーコードを設定し呼び出し元にステップ8 9 9でリターンする。

【0 1 0 6】次に、クローズプリフックルーチン5 3の処理フローの一例を、図9を参照して説明する。まず、ステップ9 1 0で、コピーファイルテーブル1 7 0 0の各ノード1 7 9 0のファイル名1 7 0 3に要求ファイル名が存在するかどうかを調べ、存在すればさらにそのノード1 7 9 0のコピーバックフラグ1 7 0 6を参照する。該参照の結果コピーバック中ならば、ステップ9 2 0で当該ノードのコピーバックフラグ1 7 0 6を監視し、コピーバック終了を待つ。

【0 1 0 7】コピーバック中でなかった場合、あるいはコピーバックが終了するとステップ9 3 0で前記コピー

バックフラグ1 7 0 6をセットする。次にステップ9 4 0でコピーバックルーチン2 5を新たなプロセスとして起動し、ステップ9 9 9で呼び出し元にリターンする。

【0 1 0 8】次に、ディレクトリ作成プリフックルーチン5 4の処理フローの一例を、図10を参照して説明する。まず、ステップ1 0 1 0で、エージェントプログラム3 0のユーザープロセスサービスルーチン3 2に対し、要求ディレクトリに対応する第二の情報処理装置1 3 0上のディレクトリの作成を依頼する。ステップ1 0 2 0では、前記作成が正常終了したかどうかを判定し、異常終了の場合にはステップ1 0 3 0でエラーコードを設定し呼び出し元にステップ1 0 9 9でリターンする。

【0 1 0 9】ディレクトリ削除プリフックルーチン5 5の処理フローの一例を図11に示す。本ルーチンは、上述のディレクトリ作成プリフックルーチン5 4とは、作成か削除かが異なるだけで、その他の処理は同様であるため、その説明は省略する。

【0 1 1 0】次に、コピーバックルーチン2 5の処理フローの一例を、図13を参照して説明する。本ルーチンがプロセスとして起動されると、まず、ステップ1 3 0 5でコピーファイルテーブル1 7 0 0の当該ノード1 7 9 0のイレース待ちフラグ1 7 0 8がセットされているかどうかを調べ、セットされていれば、以下のコピーバック処理を行わずにステップ1 3 6 0に制御を移す。また、イレース待ちフラグ1 7 0 8がセットされていなければステップ1 3 0 6で、前記ノード1 7 9 0のオープン待ちフラグ1 7 0 7がセットされているかどうかを調査する。該調査の結果、セットされていた場合は、ステップ1 3 2 0に制御を移す。

【0 1 1 1】イレース待ちフラグ1 7 0 8、オープン待ちフラグ1 7 0 7の何れもセットされていなかった場合にはステップ1 3 0 7で時間待ちを行った後、ステップ1 3 0 8で、コピーバックディレイタイムとして指定された時間が経過したかどうかを調べ、経過していなければステップ1 3 0 5に戻る。ここで、前記経過時間は、コピーバックルーチンが起動されてからの時間である。また前記ステップ1 3 0 7での時間待ちの時間は、前記コピーバックディレイタイムの1/10以下の値を設定する。

【0 1 1 2】前記コピーバックディレイタイムが経過するか、オープン待ちフラグ1 7 0 7が検出されると、ステップ1 3 2 0が実行される。該ステップ1 3 2 0では、コピーファイルテーブルのリードオープン数1 7 0 4、ライトオープン数1 7 0 5を、要求ファイルのオープンモードにしたがって減算処理を行う。即ち、リードモードだけならリードオープン数1 7 0 4から、ライトモードだけならライトオープン数1 7 0 5から、リードライトモードならリードオープン数1 7 0 4とライトオープン数1 7 0 5の両方からそれぞれ1を引いた値に設定する。

【0113】次に、ステップ1330では要求ファイルがライトオープンかどうかを判定する。続くステップ1331では、ファイルへの書き込みが行われたかどうかを、OS6bが提供する、例えばアーカイブビット等のファイル情報から判定する。ステップ1332では、最後のライトオープンであったかどうかを、前記ライトオープン数1705が0であるかどうかで判定する。以上の判定の結果、最後のライトオープンで且つファイルへの書き込みがあった場合にだけステップ1333のコピーバックを実行し、そうでなければ前記コピーバックは行わないよう動作する。前記ステップ1333のコピーバックとは、第一の情報処理装置120上に作成された要求ファイルの内容コピーを、第二の情報処理装置130に書き戻す処理をいう。前記処理は、エージェントプログラム30のユーザプロセスルーチン32を介して行う。

【0114】次に、ステップ1350で、最後のクローズかどうかを判定する。これは、前記リードオープン数1704とライトオープン数1705の両方が0となっているかどうかで容易に判定できる。また、ステップ1351ではコピーファイルテーブル1700のオープン待ちフラグ1707がセットされているかを調べる。上記判定の結果、最後のクローズで、且つオープン待ちがなければステップ1352、1355を実行する。ステップ1352では、第一の情報処理装置120上に作成された要求ファイルのファイルサイズを取得し、ステップ1355でスパースファイル化し、ファイルサイズを保持したまま内容を消去する。

【0115】ステップ1358では、コピーファイルテーブル1700から当該ノード1790を削除する。一方、ステップ1370では、コピーファイルテーブル1700の当該ノード1790のコピーバックフラグ1706をクリアする。最後に、ステップ1360で、ファイルハンドルテーブル1600から、当該ノード1690を消去し、ステップ1399で自プロセスであるコピーバックプロセスを終了する。

【0116】以上述べたように、本ルーチンによれば、コピーバックディレイタイムの期間に、同一ファイルに対するファイル削除要求が発生した場合、コピーバックを中止する。これによって、アプリケーションプログラム8aが作業用の一時ファイルを作成し、用済み後にクローズ、削除という操作を行った場合などに、不要なコピーバックを行わずに済む。

【0117】さらに、本ルーチンによれば、コピーバックディレイタイムの期間に同一ファイルに対するオープン要求が来たときは、直ちにコピーバックを開始し、該オープン処理を早く開始できるように動作する。

【0118】さらに、本ルーチンによれば、コピーバック終了までに、同一ファイルに対するファイルオープン要求が来たときは、コピーバック後のスパースファイル

化を行わないので、該ファイルオープン時には、第二のサーバ情報処理装置130からのファイルのデータコピーを行わずに済む。

【0119】次に、ディレクトリ同期ルーチン26の処理フローの一例を、図18を参照して説明する。本ルーチンにおいては、最初、ステップ1810で前回設定された同期時刻（以下、前回同期時刻と呼ぶ）を初期化し、0年0月0時0分0秒とする。次に、ステップ1820でマウントパステーブル1400の先頭ポインタ1820を取得する。ステップ1830では、取得した前記ポインタが指し示すノード1490からマウントパス1403、リモートパス1404のデータ組を取得する。

【0120】次に、ステップ1831で前回同期時刻を取得し、ステップ1832で第二の情報処理装置130のOS6cに時刻を問い合わせ、ステップ1833で前記問い合わせで得た時刻を新たな前回同期時刻として格納する。次に、ステップ1830で取得したマウントパス、リモートパス、ステップ1831で取得した前回同期時刻を引数として、ステップ1834で同期実行処理ルーチン27をコールし、一つのマウントパスに対する同期処理を実行する。

【0121】ステップ1835では、全マウントパスの同期が終了したかどうかを、参照しているマウントパステーブル1400のノード1490のポインタ1402がNULLであるかで判定する。全マウントパスの同期が終了していなければ、再びステップ1830以下を実行する。全マウントパスの同期が終了していたら、ディレクトリ同期インターバル時間で指定した時間、または停止指示のシグナルをメインルーチン21から受け取るまでステップ1840でスリープする。

【0122】ステップ1850では、前記スリープから抜けた原因が前記停止指示によるものかどうかを調べ、停止指示ならばディレクトリ同期ルーチンを終了し、停止指示でなければステップ1820以下を繰り返し実行する。

【0123】以上述べたフローによって、同期対象ディレクトリの内、第一のサーバ情報処理装置120と第二のサーバ情報処理装置130との差分だけを更新し、同期することができる。

【0124】次に、同期実行処理ルーチン27の処理フローの一例を、図19を参照して説明する。最初、ステップ1910では、本ルーチンへ引数として渡された前回同期時刻とリモートパスを、第二の情報処理装置上のエージェントプログラム30のディレクトリ同期サービスルーチン33に渡し、リモートパスの下位階層部分のファイル（ディレクトリファイルを含む）で前回同期時刻以降に更新されたものの名称と、ファイルサイズ、ファイル日付の一覧を変更ファイルリストとして受け取る。

【0125】ステップ1920以下では、前記リストのファイル、ディレクトリを一個毎に処理する。まず、ステップ1920でディレクトリファイルかどうかを判定し、ディレクトリファイルであった場合には、ステップ1960でサブディレクトリ同期ルーチン28をコールし、そのディレクトリの下位の階層部分全ての同期をとる。一方、データファイルであった場合には、ステップ1930で、第一の情報処理装置上120上の対応するマウントパスに既にそのファイルが存在するかを調べ、存在しなければステップ1940を、存在した場合は1935を実行する。

【0126】ステップ1935では、コピーファイルテーブル1700に当該ファイルが登録されているかどうかを検索することによって、該ファイルが使用中かどうかを判定する。コピーファイルテーブル1700に当該ファイルが登録されていた場合には、該ファイルは使用中であるので、ステップ1940のスパースファイル作成をスキップする。ステップ1940では、ファイル名、サイズ、日付を引数としてスパースファイル作成ルーチン60をコールし、当該ファイルの内容を消去すると共に、ファイルサイズ、ファイル日付を設定する。

【0127】ステップ1950では、ステップ1910で取得した更新ファイル一覧の全てのファイル同期を終了したかどうかを判定し、終了するまでステップ1920以下を繰り返す。全ファイルの同期が終了するとステップ1999で呼び出し元へリターンする。

【0128】次に、上記同期実行処理ルーチン27から呼ばれるサブディレクトリ同期ルーチン28の処理フローを、図20を参照して説明する。ステップ2010では、引数として受け取ったディレクトリの直下のファイルリストを、第二の情報処理装置上のエージェントプログラム30のディレクトリ同期サービスルーチン33から得る。このリストには、各ファイル、ディレクトリの名称、日付、サイズの情報が含まれる。

【0129】ステップ2020以下では、前記ファイルリストのファイル、ディレクトリを一個ずつ処理する。ステップ2020では、ディレクトリファイルかどうかを判定し、ディレクトリであった場合には、ステップ2070でサブディレクトリ同期ルーチン28を再帰的に呼び出す。ファイルであった場合には、ステップ2025で、第一の情報処理装置上120上の対応するマウントパスに既にそのファイルが存在するかを調べ、存在しなければステップ2030を、存在した場合は2026を実行する。

【0130】ステップ2026では、コピーファイルテーブル1700に当該ファイルが登録されているかどうかを検索することによって、該ファイルが使用中かどうかを判定する。コピーファイルテーブル1700に当該ファイルが登録されていた場合には、該ファイルは使用中であるので、ステップ2030のスパースファイル作

成をスキップする。使用中でなかった場合には、ステップ2030でスパースファイル作成ルーチン60をファイル名称、サイズ、日付を引数としてコールし、当該ファイルの内容を消去すると共に、ファイルサイズ、ファイル日付を設定する。

【0131】ステップ2040では、ステップ2010で取得したファイルディレクトリ一覧の全てのファイル同期を終了したかどうかを判定し、終了するまでステップ2020以下を繰り返す。全ファイルの同期が終了すると、ステップ2050でファイルディレクトリリストに存在しないファイルまたはディレクトリが、ローカル側の当該ディレクトリ内に存在しないかを調べ、存在した場合にはこれらを全て削除する。以上の処理を終えると、ステップ2099で呼び出し元へリターンする。

【0132】次に、スパースファイル作成ルーチン60の処理フローの一例を、図12を参照して説明する。最初、ステップ1205で、引数として受け取ったファイル名のファイルが存在するかを確認し、存在すればステップ1210で削除する。次にステップ1220で新たにファイルを作成する。ステップ1225では、引数として受け取ったファイルサイズ分ファイルポインタをずらして、ステップ1230で1バイトのダミーデータを書き込む。そして、ステップ1235で該ファイルをクローズする。

【0133】ステップ1240で、該ファイルの日付を引数として受け取ったものに変更し、ステップ1245でファイルの属性情報の一つである変更ビットをクリアする。以上でスパースファイルの作成を完了し、ステップ1299で呼び出し元へリターンする。

【0134】次に、エージェントプログラム30に含まれる各ルーチンの処理フローを説明する。

【0135】最初に、上述したゲートウェイプログラム20のユーザプロセスルーチン23からのログイン要求を受けつけるリモートログインサービスルーチン31について、図26を参照して説明する。

【0136】本処理において、ステップ2601では、前記ユーザプロセスルーチン23からのログイン要求あるいはサービス停止指示のイベントを監視し待つ。該イベントが発生するとステップ2602でログイン要求かどうかを判定し、ログイン要求の場合は、ステップ2603でユーザプロセスサービスルーチン32を新たなプロセスとして起動し、ここからログインを実行する。

【0137】一方、ステップ2602でログインではないと判定した場合は、発生したイベントがサービス停止指示かどうかを判定し、サービス停止指示であった場合には、ステップ2699へ進み、本処理を終了する。前記サービス停止指示以外のイベントであった場合には、ステップ2601へ戻り、イベント待ちを実行する。

【0138】次に、上記リモートログインサービスルー

チン 31 のステップ 2603 で生成、起動され、前記ゲートウェイプログラム 20 の各ルーチンからリダイレクトされた各種イベントにตอบสนองする、ユーザープロセスサービスルーチン 32 の処理フローを、図 27 を参照して説明する。

【0139】本処理は、上記リモートログインサービスルーチン 31 のステップ 2603 で起動された後、ステップ 2701 でイベント待ちを行う。ここで、本ルーチンにリダイレクトされるイベントとしては、前記ファイルアクセスフックルーチン 24 からの各種ファイルアクセス要求や、前記リモートログインルーチン 22 からのログイン、ログアウト要求等がある。

【0140】ステップ 2702～2710 までの判定処理では、該イベントが、ログイン要求、ログアウト要求、ファイルオープン要求、ファイルクローズ要求、ファイルリード要求、ファイルライト要求、ファイル削除要求、ディレクトリ作成要求、及び、ディレクトリ削除要求のいずれに該当するかを判定し、該判定結果に応じて、ステップ 2731～2739 のうち該当する処理を実行する。

【0141】次に、ステップ 2740 では、前記ユーザープロセスルーチン 23 に、前記リモートログインサービスルーチン 31 から引き継いだソケットを用いて、各種要求に対する処理結果を伝えるためのリターンコードを設定する。ステップ 2741 では、該リターンコードを含む処理の実行結果を、前記ソケットを通して、第一のサーバ情報処理装置 120 へ送信して前記イベントの発行元へ伝え、その後、またステップ 2701 へ戻りイベント待ちを行う。

【0142】上記ステップ 2702～2710 で認識されるアクセス要求以外のイベントの場合には、該イベントがサービス停止指示かどうかを判定し、サービス停止指示であった場合には、ステップ 2799 へ進み、本ルーチンを終了する。前記サービス停止指示以外のイベントであった場合には、ステップ 2701 へ戻り、イベント待ちを実行する。

【0143】次に、ディレクトリ同期サービスルーチン 33 について、図 28 を参照して説明する。本ルーチンは、前記ゲートウェイプログラム 20 のディレクトリ同期ルーチン 26 からの要求に応じ、同期実行処理ルーチン 27 のステップ 1910 の処理に対応して、第二のサーバ情報処理装置 130 のディレクトリ構造に関する変更ファイルリストを作成し、該作成したリストを返す処理である。

【0144】本処理では、ステップ 2801 でイベント待ちをして、ステップ 2802 で該イベントが前記同期実行処理ルーチン 27 のステップ 1910 からの更新ファイル取得指示かどうかを判定する。該判定の結果、それが更新ファイル取得指示である場合には、ステップ 2803 で更新ファイルリストを初期化し、ステップ 28

04～2807 で、該更新ファイル取得指示に引数として含まれている前同期時刻と、マウントパステーブル 1400 のリモートパス 1404 とに基づいて、更新ファイルリストを作成する。

【0145】すなわち、ステップ 2804 で、前記引数に含まれるリモートパス 1404 の下位層側に位置する、第二のサーバ情報処理装置 130 のディレクトリ構造に含まれるファイル及びサブディレクトリファイルを順次検索し、ステップ 2805 で、各ファイルについて前記同期時刻以降に変更があったかどうかを判定し、該判定の結果、変更があった場合には、ステップ 2806 で、該当するファイルの名称を前記変更ファイルリストに追加する。ステップ 2805 で変更がないと判定された場合にはステップ 2807 へ進み、ステップ 2807 で、検索の対象とされているファイルのすべてについて検索が終了したかどうかを判定し、終了まで上記処理を繰り返す。

【0146】ステップ 2808 では、ステップ 2804～2807 の処理の結果得られた変更ファイルリストを第一のサーバ情報処理装置 120 へ送信して、前記同期実行処理ルーチン 27 へ返す。

【0147】上記イベントが更新ファイル取得指示以外のイベントの場合には、該イベントがサービス停止指示かどうかを判定し、サービス停止指示であった場合には、ステップ 2899 へ進み、本ルーチンを終了する。前記サービス停止指示以外のイベントであった場合には、ステップ 2801 へ戻り、イベント待ちを実行する。

【0148】

【発明の効果】本発明によれば、異なるファイルサーバプログラムを備えた複数のサーバ情報処理装置上のファイルに対してアクセスする場合でも、単一のファイルサーバプログラムを備えたサーバ情報処理装置上のファイルへアクセスする場合とほぼ同様な操作性を確保できる、異種ファイルへのアクセスを可能とする情報処理システム及びその制御方法を提供することができる。

【0149】さらに、本発明は、第二のサーバ情報処理装置上のディレクトリ構造を第一のサーバ情報処理装置上に定期的にコピーすることにより、第一の情報処理装置用のファイル一覧表示プログラムがそのまま利用できるという効果がある。

【0150】さらに、本発明は、第二のサーバ情報処理装置のディレクトリ構造を第一のサーバ情報処理装置上に定期的にコピーする際に実行する前記第一のサーバ情報処理装置上へのファイル作成において、ディレクトリファイル以外の通常のファイルを、中身の無いスパースファイルとしてファイルを作成することにより、前記ファイルが使用する第一の情報処理装置の磁気ディスク容量が少なく済むという効果がある。

【0151】さらに、本発明は、前回のディレクトリ同

期実行時刻以後に変更されたファイルだけを同期処理の対象とするので、第二のサーバ情報処理装置から第一のサーバ情報処理装置に送る情報量が少なく済み、処理の高速化、LANトラフィックの低減を図れるという効果がある。

【0152】さらに、本発明によれば、複数ユーザーが同時に同一ファイルを使用する際にも無駄なコピー処理が発生しない。また、複数のユーザーが使用中であったり、クローズに続いて再度オープンされる場合には、当該ファイルをスパースファイル化せず、コピーファイルを引き続き利用できる。また、クローズに続いて削除される場合に、不要なコピーバックをせずに済む。また、コピーバック手段は、別プロセスとしてバックグラウンドで動作するので、前記コピーバック手段の完了を待つことなくクライアント情報処理装置からのクローズ要求は終了するので、直ちに次の処理を行うことができる。

【0153】よって、本発明は、クライアント情報処理装置から出された第二のサーバ情報処理装置上のファイルに対するアクセス処理の高速化と共に、LANトラフィックの低減を図れるという効果がある。

【0154】

【図面の簡単な説明】

【図1】本発明を適用した異種ファイルアクセス手段を備えた情報処理システムの一実施例の構成を示すブロック図である。

【図2】図1の実施例におけるゲートウェイプログラム20のメインルーチン21の処理フローの一例を示すフローチャートである。

【図3】図1の実施例におけるゲートウェイプログラム20のリモートログインルーチン22の処理フローの一例を示すフローチャートである。

【図4】図1の実施例におけるゲートウェイプログラム20のユーザープロセスルーチン23の処理フローの一例を示すフローチャートである。

【図5】図1の実施例におけるゲートウェイプログラム20のファイルアクセスフックルーチン24の処理フローの一例を示すフローチャートである。

【図6】図1の実施例におけるゲートウェイプログラム20のオープンプリフックルーチン50の処理フローの一例を示すフローチャートである。

【図7】図1の実施例におけるゲートウェイプログラム20のオープンポストフックルーチン51の処理フローの一例を示すフローチャートである。

【図8】図1の実施例におけるゲートウェイプログラム20のイレースプリフックルーチン52の処理フローの一例を示すフローチャートである。

【図9】図1の実施例におけるゲートウェイプログラム20のクローズプリフックルーチン53の処理フローの一例を示すフローチャートである。

【図10】図1の実施例におけるゲートウェイプログラ

ム20のディレクトリ作成プリフックルーチン54の処理フローの一例を示すフローチャートである。

【図11】図1の実施例におけるゲートウェイプログラム20のディレクトリ削除プリフックルーチン55の処理フローの一例を示すフローチャートである。

【図12】図1の実施例におけるゲートウェイプログラム20のスパースファイル作成ルーチン60の処理フローの一例を示すフローチャートである。

【図13】図1の実施例におけるゲートウェイプログラム20のコピーバックルーチン25の処理フローの一例を示すフローチャートである。

【図14】図1の実施例におけるマウントパステーブル1400の一例を示す説明図である。

【図15】図1の実施例におけるユーザーコネクションテーブル1500の一例を示す説明図である。

【図16】図1の実施例におけるファイルハンドルテーブル1600の一例を示す説明図である。

【図17】図1の実施例におけるコピーファイルテーブル1700の一例を示す説明図である。

【図18】図1の実施例におけるゲートウェイプログラム20のディレクトリ同期ルーチン26の処理フローの一例を示すフローチャートである。

【図19】図1の実施例におけるゲートウェイプログラム20の同期実行処理ルーチン27の処理フローの一例を示すフローチャートである。

【図20】図1の実施例におけるゲートウェイプログラム20のサブディレクトリ同期ルーチン28の処理フローの一例を示すフローチャートである。

【図21】図1の実施例における第一のサーバ情報処理装置120のディレクトリ構造の一例を示す説明図である。

【図22】図1の実施例における第二のサーバ情報処理装置130のディレクトリ構造の一例を示す説明図である。

【図23】図1の実施例におけるゲートウェイプログラム20稼働時の第一のサーバ情報処理装置120のディレクトリ構造の一例を示す説明図である。

【図24】図1の実施例におけるゲートウェイパラメータ設定ファイル199の一例を示す説明図である。

【図25】図1の実施例におけるスパースファイルの構造を示す説明図である。

【図26】図1の実施例におけるエージェントプログラム30のリモートログインサービスルーチン31の処理フローの一例を示すフローチャートである。

【図27】図1の実施例におけるエージェントプログラム30のユーザープロセスサービスルーチン32の処理フローの一例を示すフローチャートである。

【図28】図1の実施例におけるエージェントプログラム30のディレクトリ同期サービスルーチン33の処理フローの一例を示すフローチャートである。

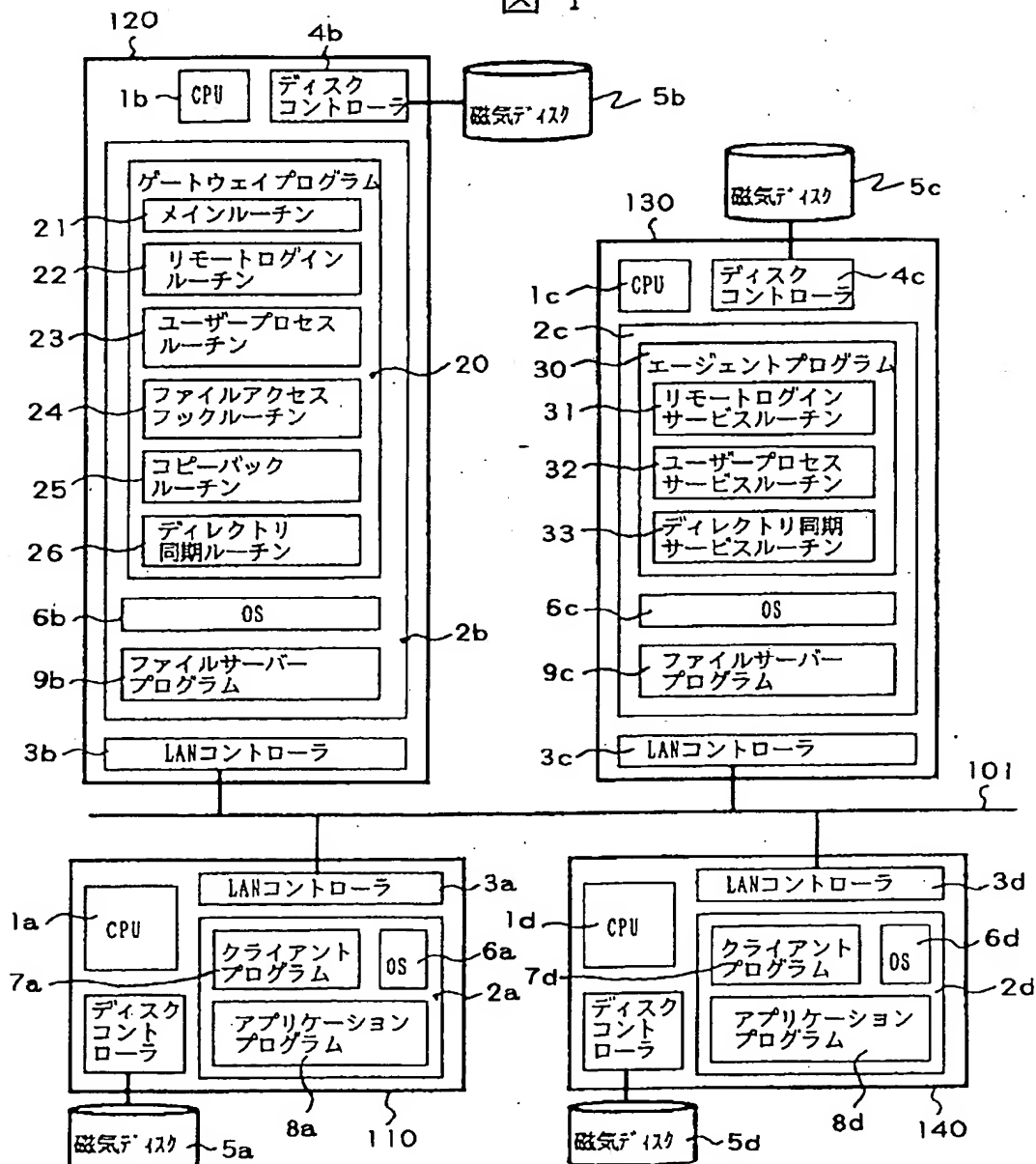
【符号の説明】

1…CPU, 2…主メモリ, 3…LANコントローラ,
4…ディスクコントローラ, 5…磁気ディスク, 6…OS,
7…クライアントプログラム, 8…アプリケーション
プログラム, 9…ファイルサーバプログラム, 20…*

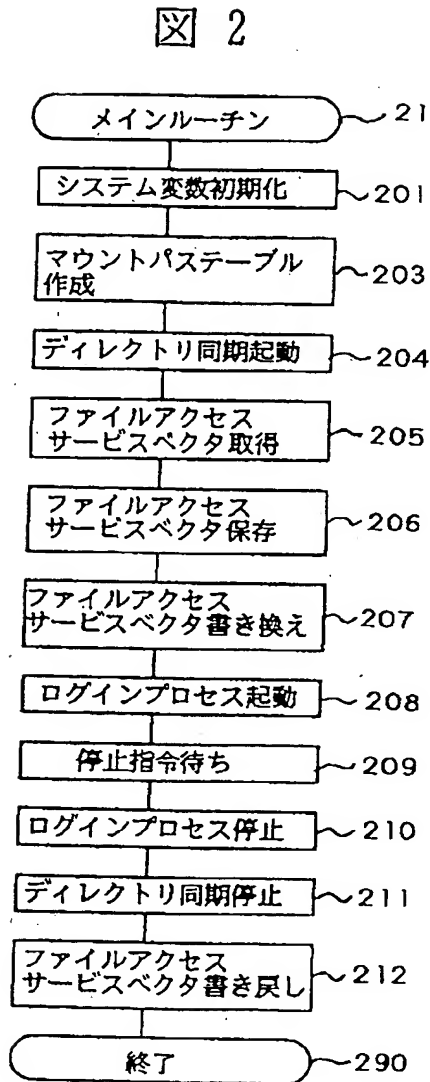
*ゲートウェイプログラム, 30…エージェントプログラ
ム, 101…LAN, 110…第一のクライアント情報
処理装置, 120…第一のサーバ情報処理装置, 130
…第二のサーバ情報処理装置, 140…第二のクライ
アント情報処理装置。

【図1】

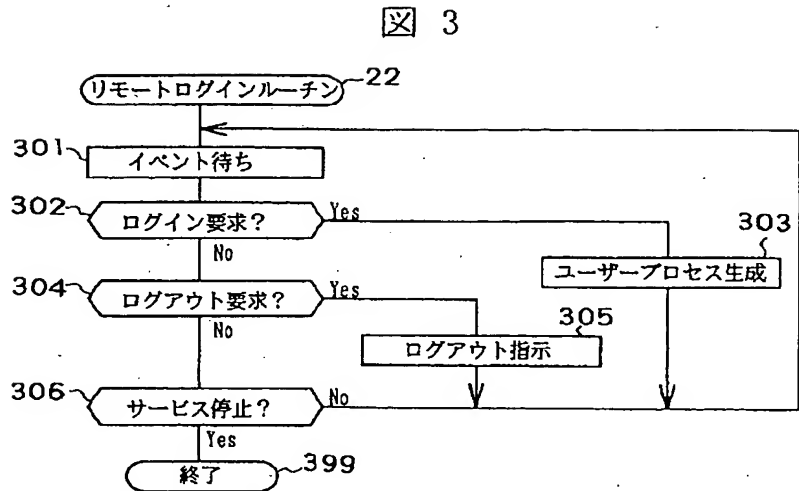
図 1



【図 2】

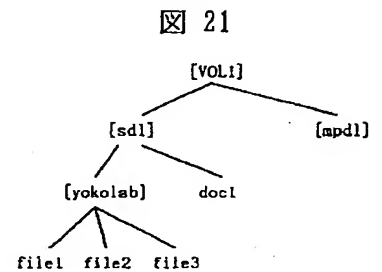
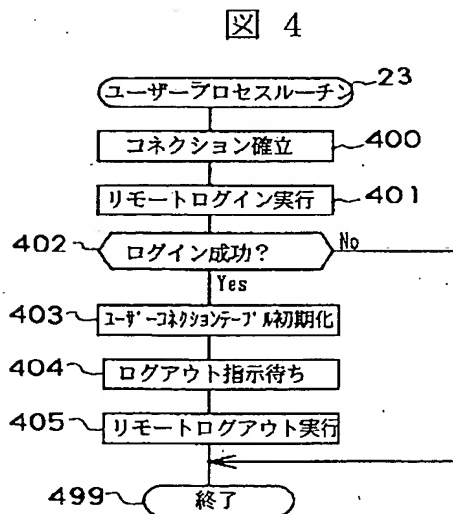


【図 3】



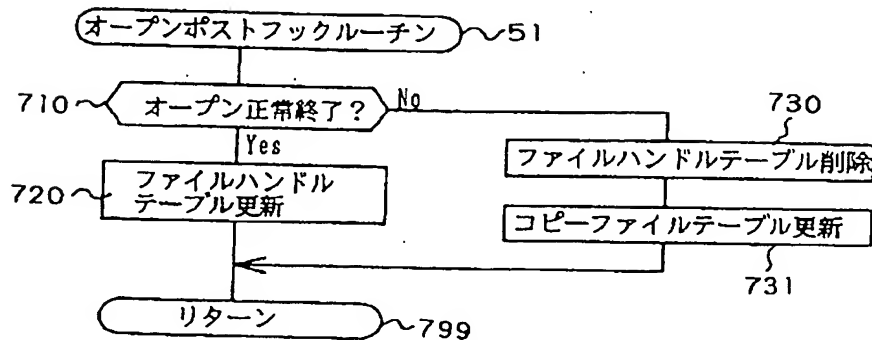
【図 4】

【図 21】



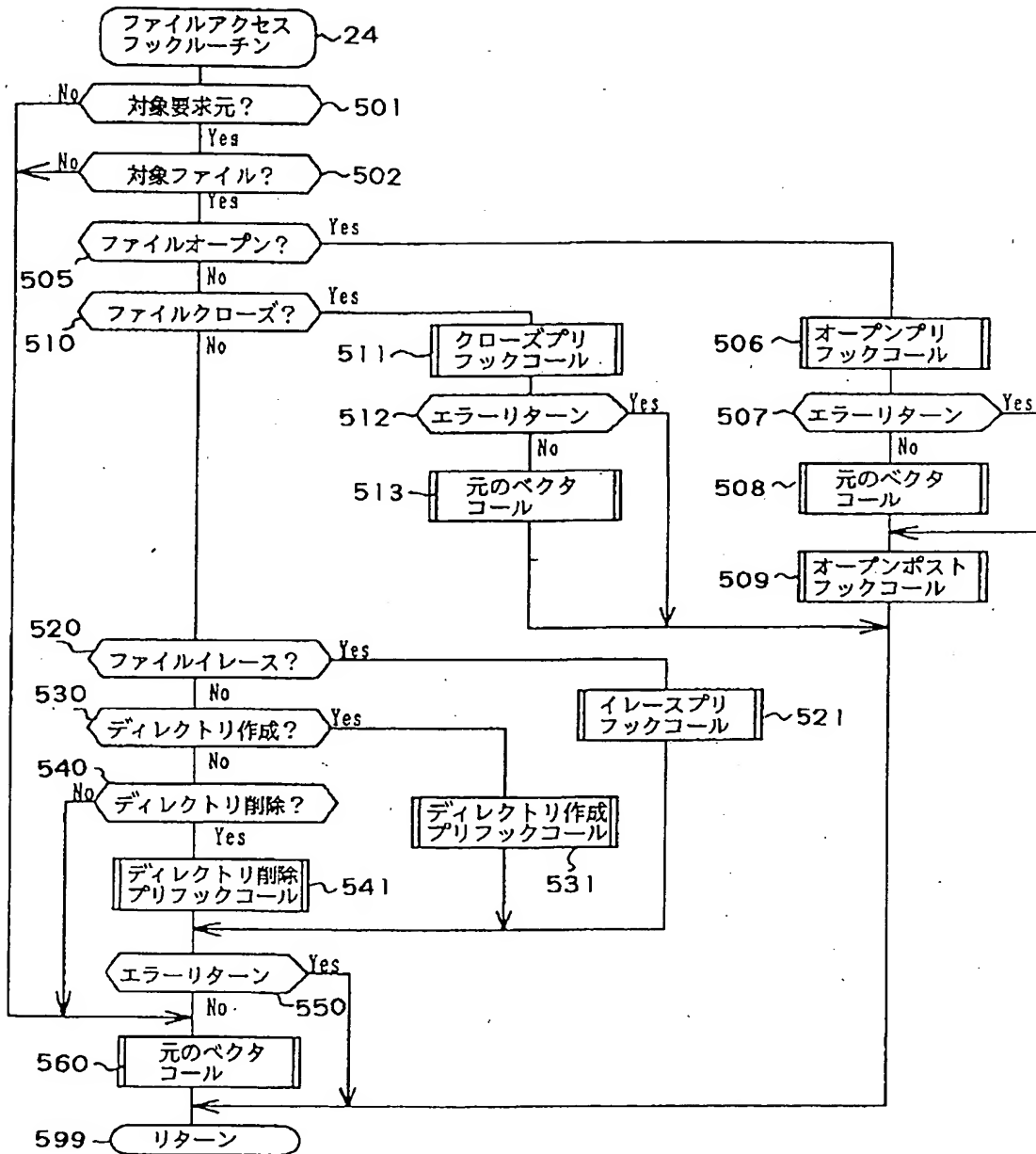
【図 7】

図 7



【図 5】

図 5



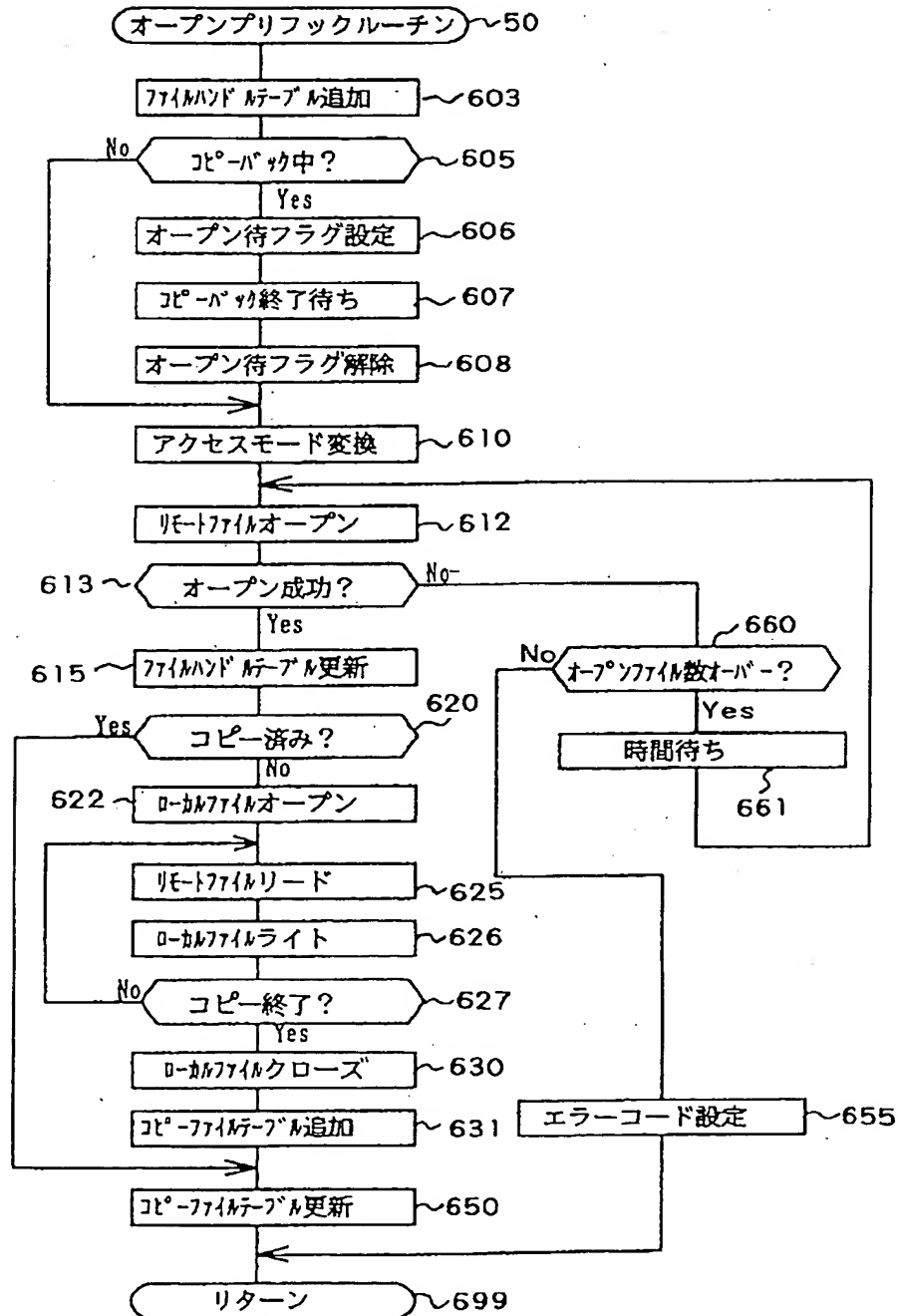
【図 24】

図 24

2410 MOUNT VOL1/mpd1 = SRV2:research/hard
 2420 COPYBACKDELAYTIME = 500
 2430 DIRSYNCINTERVAL = 6000

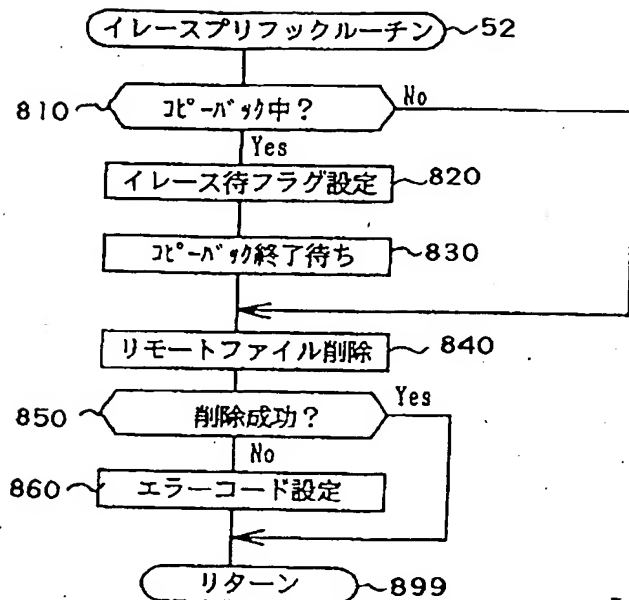
【図6】

図 6



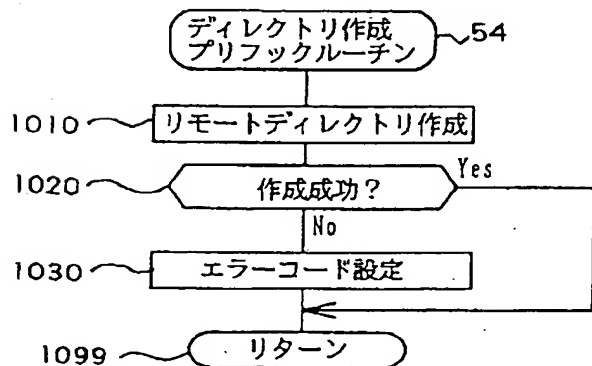
【図 8】

図 8



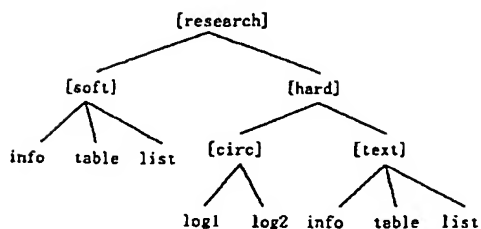
【図 10】

図 10



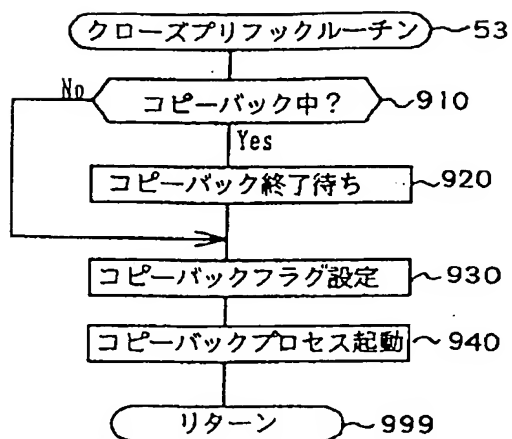
【図 22】

図 22



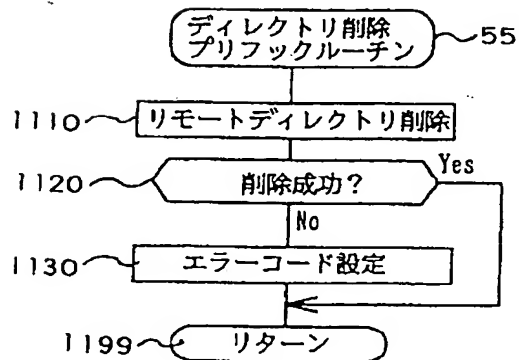
【図 9】

図 9



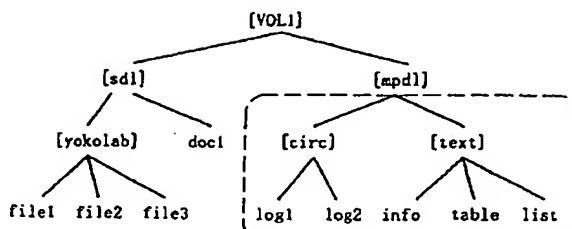
【図 11】

図 11



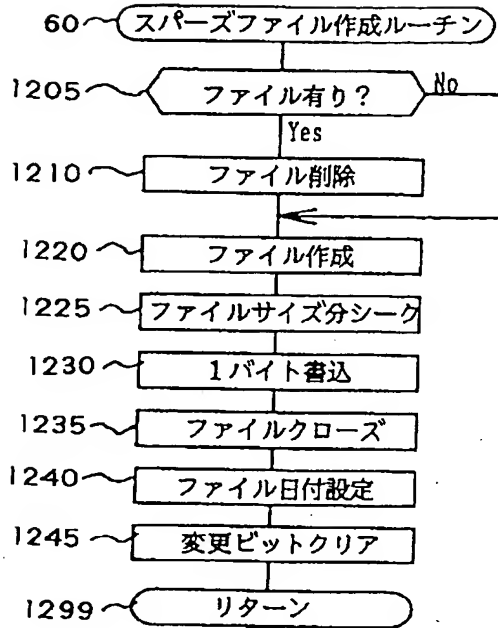
【図 23】

図 23



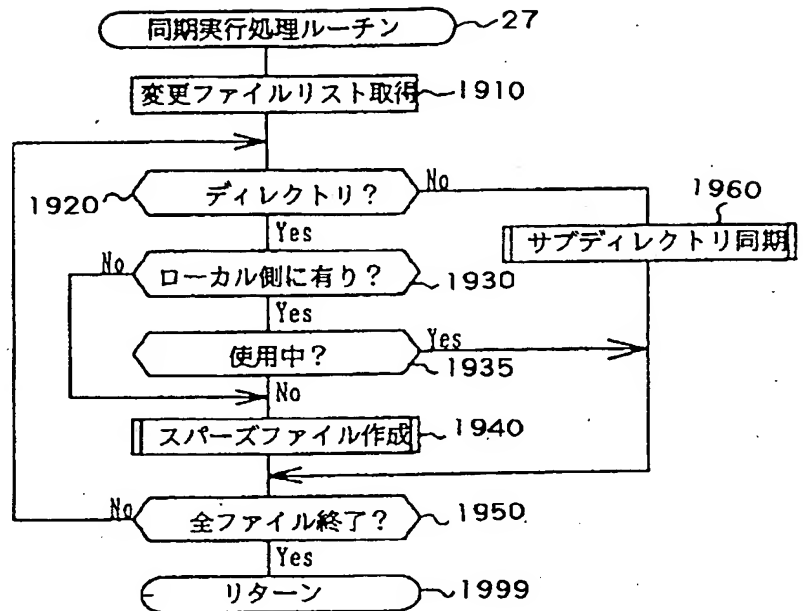
【図12】

図 12



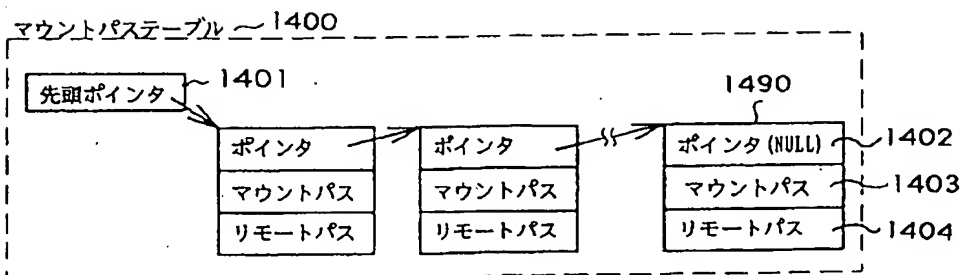
【図19】

図 19



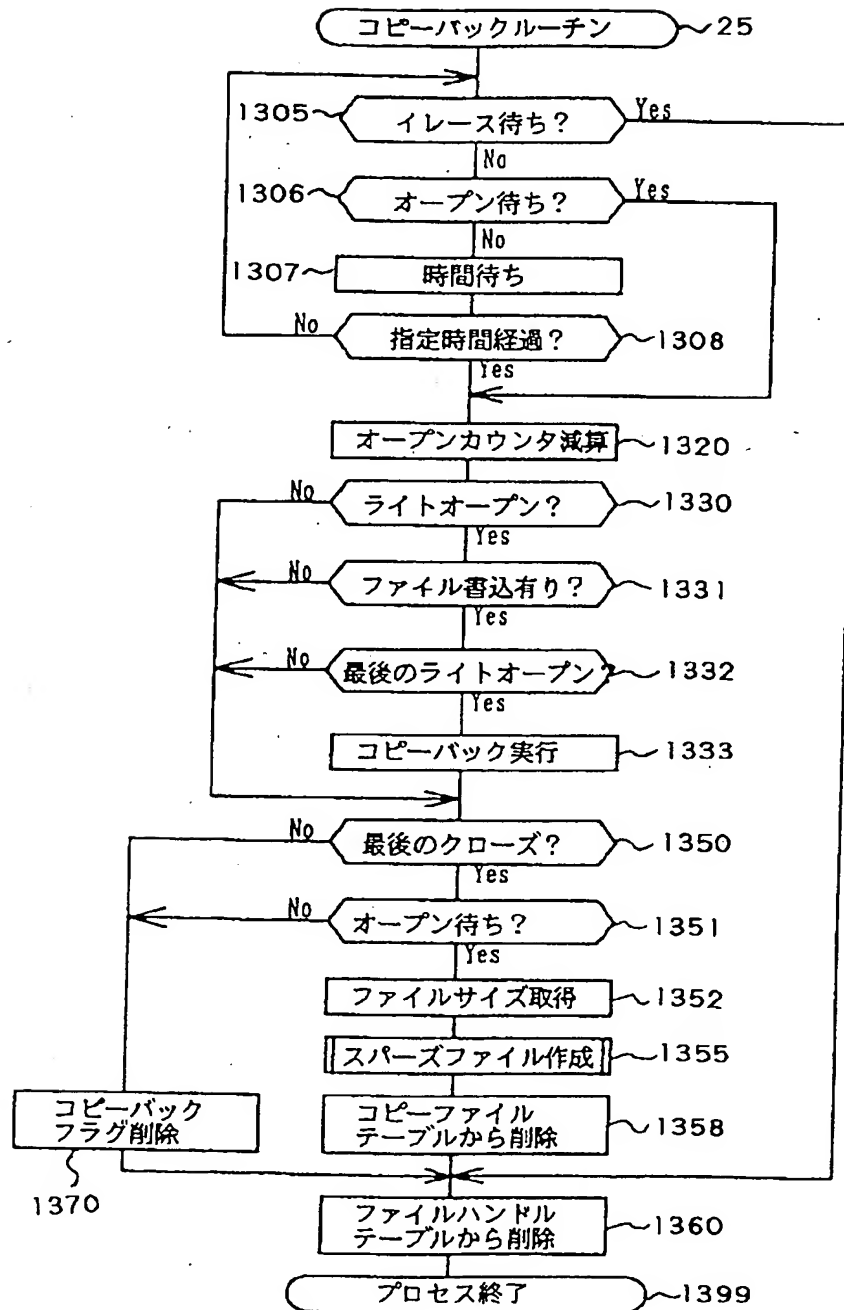
【図14】

図 14



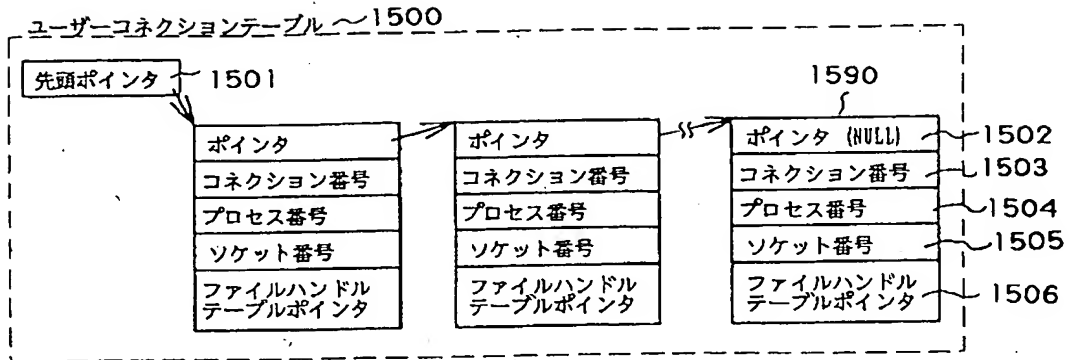
【図13】

図 13



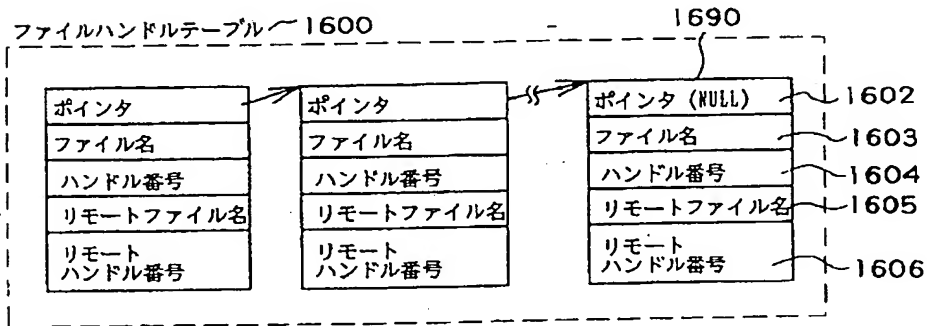
【図 15】

図 15



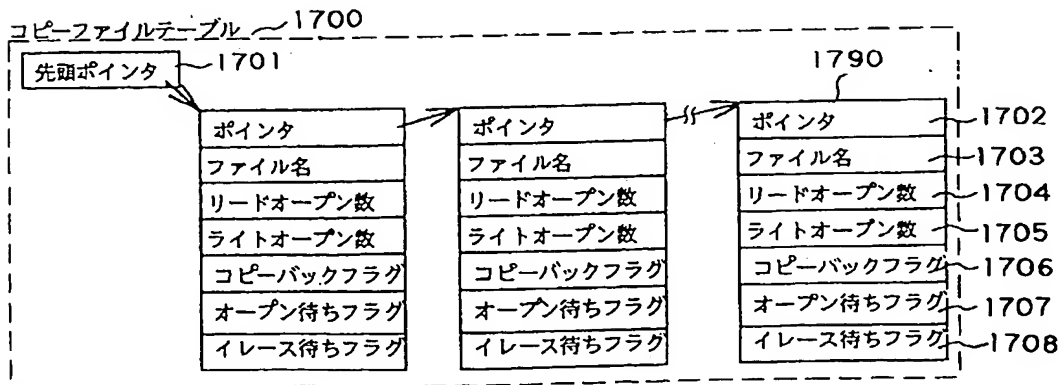
【図 16】

図 16



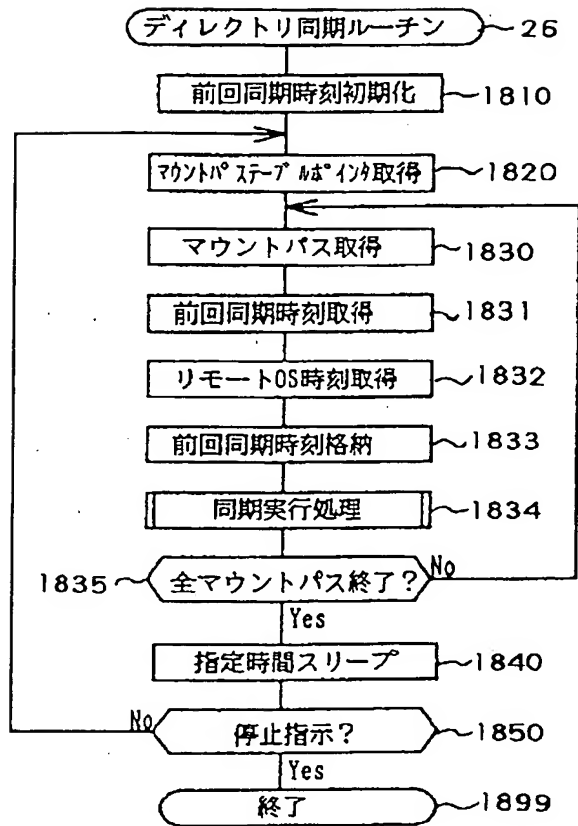
【図 17】

図 17



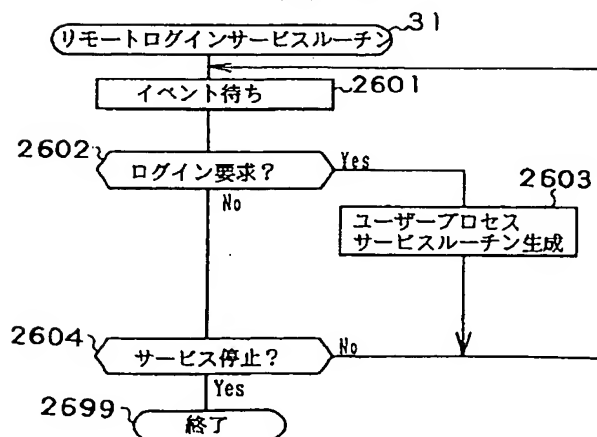
【図 18】

図 18



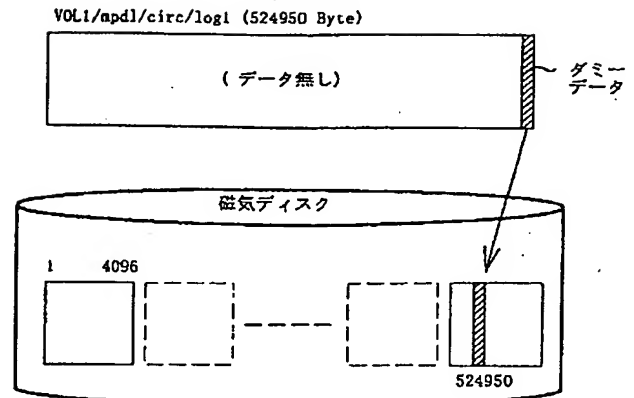
【図 26】

図 26



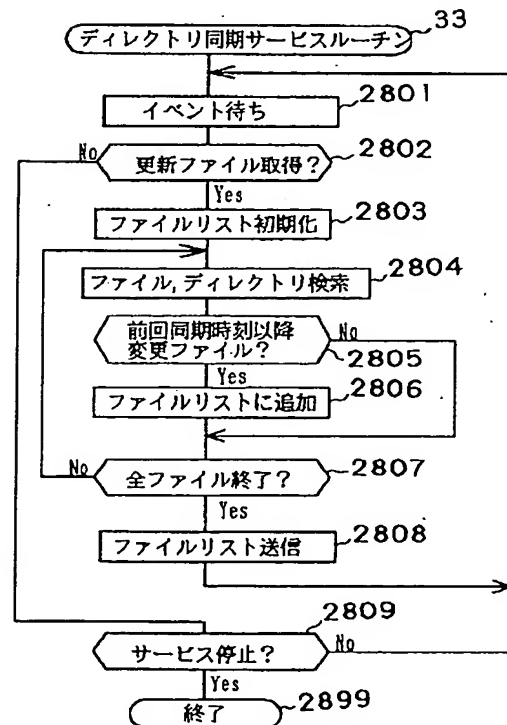
【図 25】

図 25



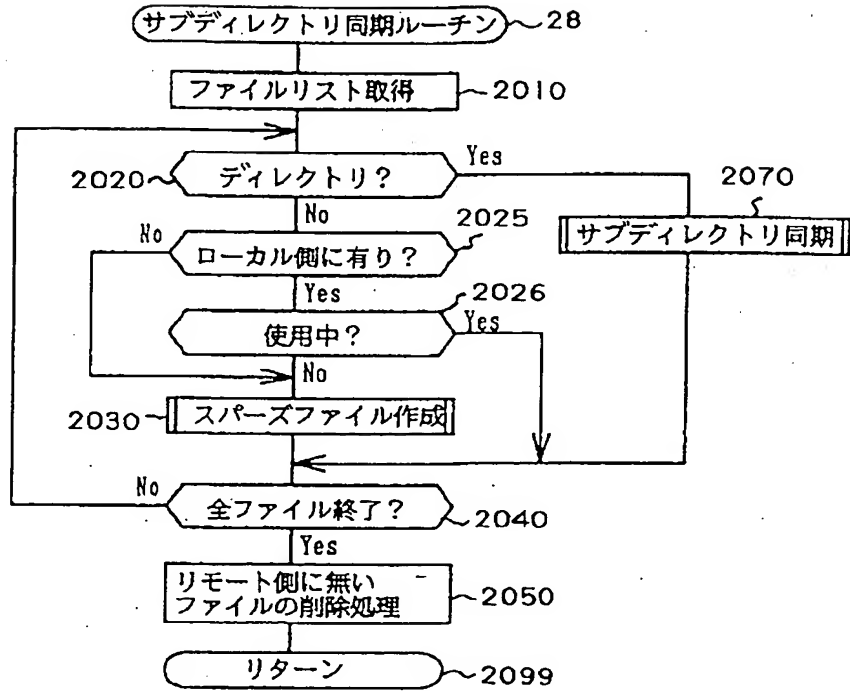
【図 28】

図 28



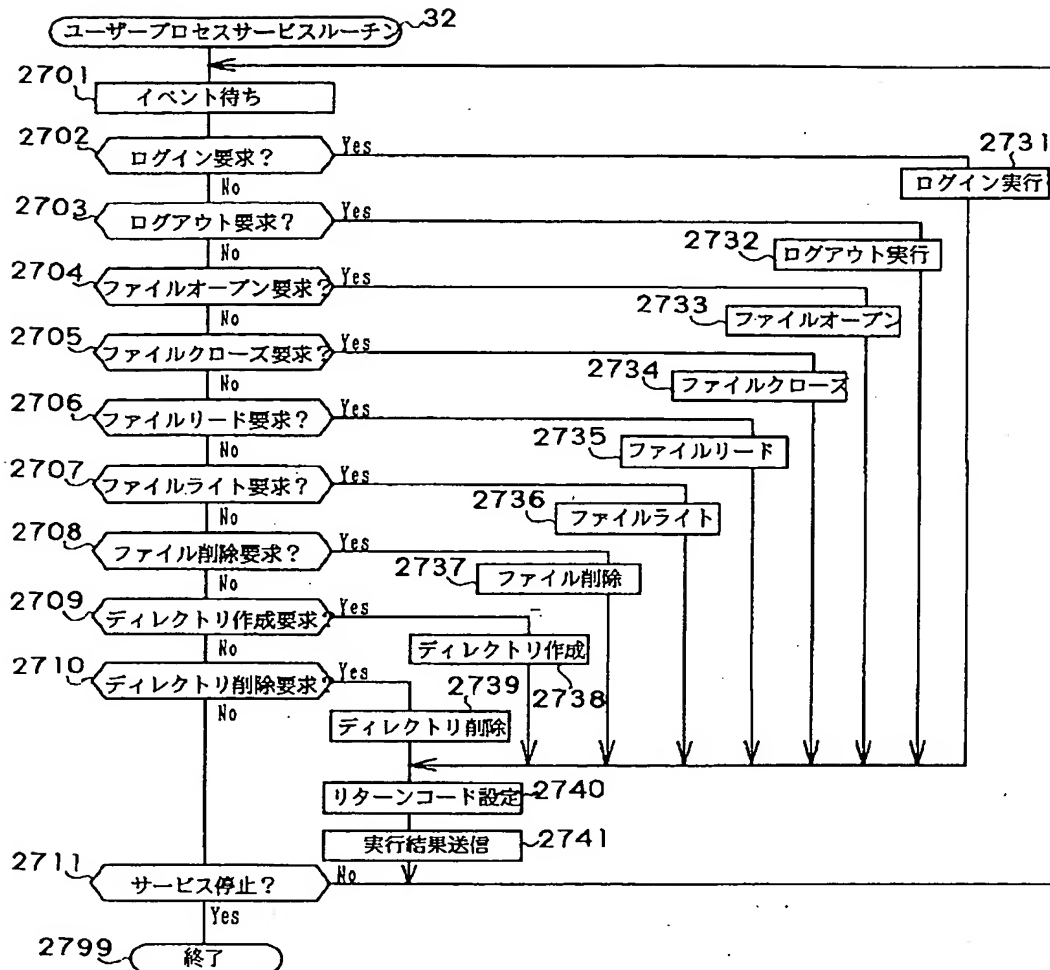
【図20】

図 20



【図27】

図 27



フロントページの続き

(72)発明者 荒井 正人
 神奈川県川崎市麻生区王禅寺1099番地 株
 式会社日立製作所システム開発研究所内

(72)発明者 中田 幸男
 神奈川県川崎市麻生区王禅寺1099番地 株
 式会社日立製作所システム開発研究所内

(72)発明者 伊藤 寿哉
 神奈川県横浜市戸塚区戸塚町5030番地 株
 式会社日立製作所ソフトウェア開発本部内

(72)発明者 森 充
 神奈川県横浜市中区尾上町6丁目81番地
 日立ソフトウェアエンジニアリング株式会
 社内